Entered ✓

951050?

INTERACTIVE SELECTION OF AMBIGUOUS 2-D SHAPES

by

ANURAG GUPTA

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

January, 1997

INTERACTIVE SELECTION OF AMBIGUOUS 2-D SHAPES

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

Anurag Gupta

to the

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

January 1997

Certificate

It is certified that the work contained in the thesis entitled **INTERACTIVE SELECTION OF AMBIGUOUS 2-D SHAPES**, by LT. Anurag Gupta, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

January 1997

Amitabha Mukerjee
Assistant Professor
Department of Mechanical Engineering
I.I.T. Kanpur

Abstract

In this work, we develop an interactive conceptual design editor with the following features :

1. Component shapes are not unique but vary within a shape class, defined as a range of shape parameters (A design sketch is an example of a shape class).
2. A large number of these shapes are displayed to the user as opposed to a single geometry. This gives the user an idea of the ambiguity inherent in the shape class.
3. The shapes can be optimized based on pre-defined criteria, or **interactively** by direct selection or by changing the shape class itself.

The principal contribution of this work is developing the interactive optimization interface. The language for specifying optimization constraints may pose its own limits on the constraints that can be specified. A looser form of interaction would permit the user to say merely that "Ah – this is just what I want", or "I do not like this", without having to quantify the reasons as an objective function. Furthermore, in the process of reviewing candidate shapes the user may be able to revise the shape domain itself by changing the range of variation of the shape parameters. These are possible only with an interactive framework.

This work draws on the axis model as used in reconstructable ambiguous shape modeling [2]. The model uses a qualitative parametrization based on medial axis or generalized Voronoi model which can be perturbed to generate a reconstructable set of shapes. We present some empirical results of this work for several cases of simple shapes.

Dedicated To

My Family Members

Acknowledgements

I feel extremely happy to express my deep and sincere gratitude towards Dr. Amitabha Mukerjee, my thesis supervisor, for his encouragement, guidance and uneasing enthusiasm throughout this thesis work. The open atmosphere for approaching him, exchanging views with him, coupled with his encouragement through thick and thin, was the main driving force throughout this work.

I also wish to express my gratitude towards Dr. Kalyanmoy Deb, Asst. Prof., Mech. Dept., for taking keen interest in my work and guiding me, especially through the problems related to Genetic Algorithms.

It is with great pleasure that I thank my friends Vikas Saxena and Atul Gupta, for their help and ideas throughout this work.

Last, but not the least, I am extremely grateful to my wife for looking after our daughter single- handedly and helping me to successfully avail of this opportunity of doing my Master's degree.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> Computer Aided Design is now embarked on a search for the alchemy that will transform it from a pedestrian tool embodying the form of design into one that will capture its functionality, the intent, the very dream from the designer's mind [10].

One of the tenets of CAD, as put forward by Requicha [13], has been "unambiguousness" i.e., a single model represents only a single object. According to Requicha, a representation scheme is a relation between solids (in real world) and their representations (models). A representation is *unambiguous* if it corresponds to a single object, and *ambiguous*, if it corresponds to several objects (Fig.1.1).

Until recently, most CAD models were unambiguous i.e., represented by an exact boundary. However, in areas such as conceptual design, object recognition, shape approximation etc. there is a need for models that are less precise. While parametric modeling in its broadest sense can describe any set of parameters related to any model of shape, in practice, the geometry is defined using a small set of parameters. The question we address is how the postulated design is to be represented particularly in the aspect of geometry. Typically, designs are represented by a set of parameters and new designs are postulated by moving to a new set of parameters in the parameter space. However, the parameters impose considerable constraints on the design space and the knowledge-based approaches mistakenly assume the parameter space to be equivalent to the design space (Fig.1.2).

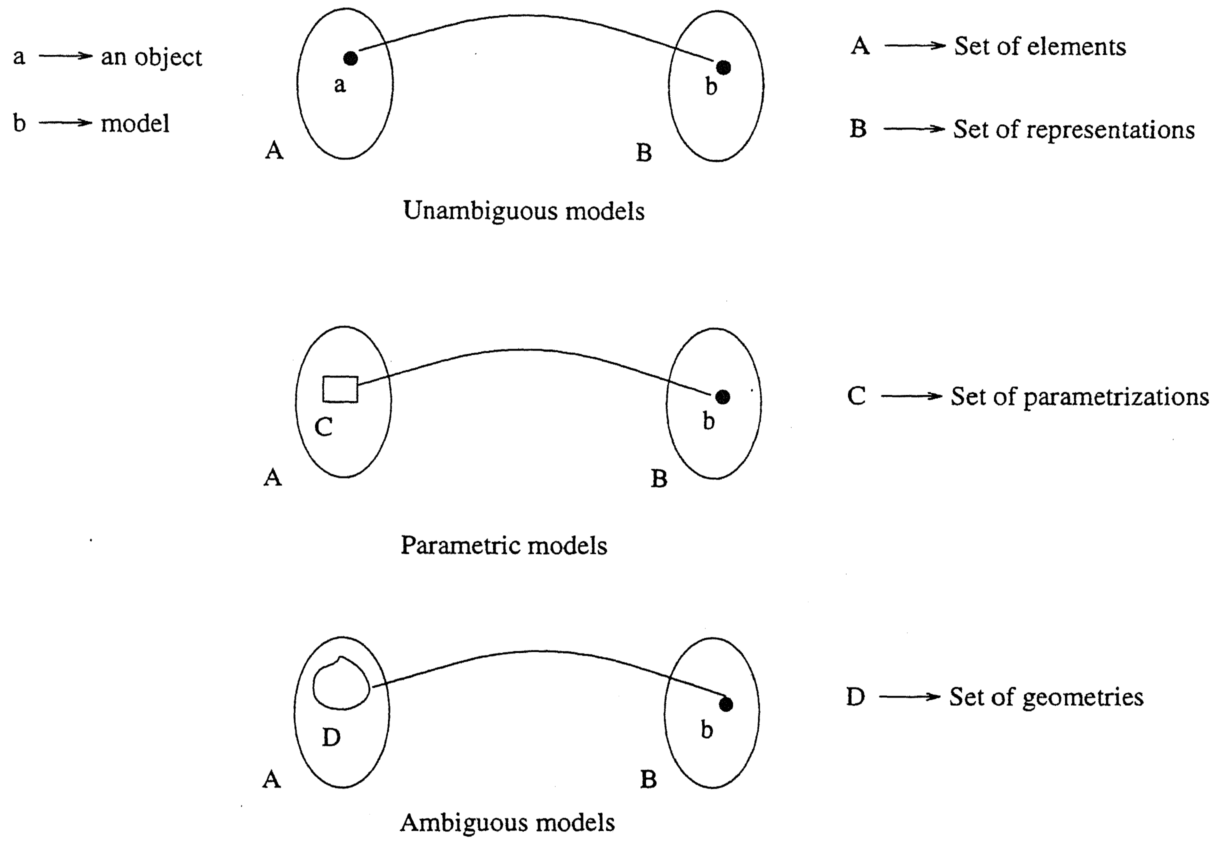


Figure 1.1: Representation Schemes.

As an example, consider a model for elongated objects. A parameter would use the notion of an aspect ratio (Fig.1.3(a)).

This permits objects such as those in Fig.1.3(b), (c) and (d), whereas what we meant by "elongated" objects may be like those shown in Fig.1.3(e), (f), (g) and (h). Thus, a shape parameter like aspect ratio fails to capture the notion of a "filled-in" shape or any other such shape class. On the other hand, a 2-parameter rectangular model is too specialized and Fig.1.3(e), (f) and (h) would fall outside the purview of this model.

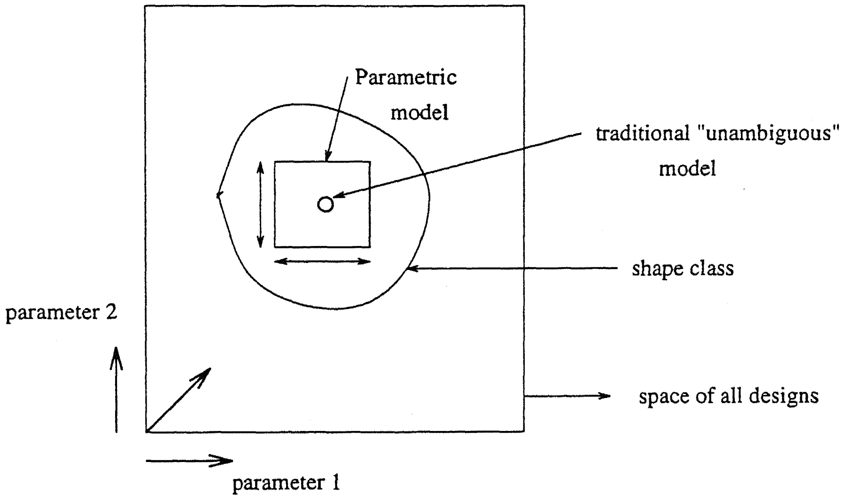


Figure 1.2: The space of all designs.

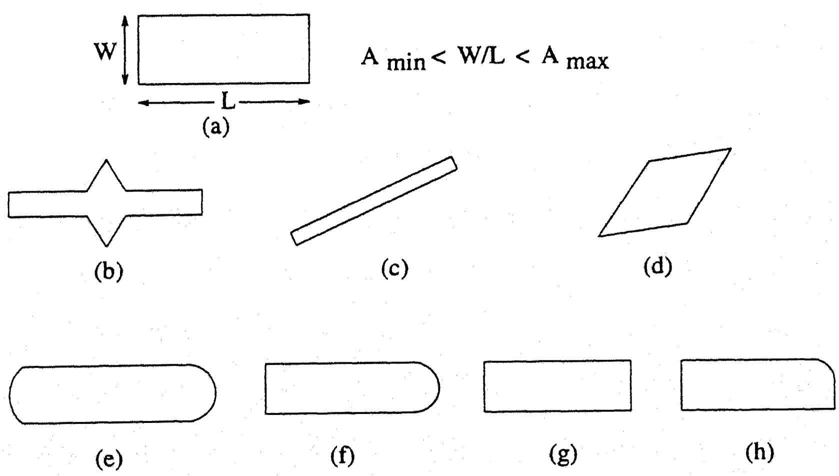


Figure 1.3: Model for elongated objects.

1.1 The Design Process

Design is directed towards an ill-defined idea of a desired state in which certain characteristics are evident. In CAD models, this ill-defined objective cannot be handled. However, recent work using AI have attempted to model design as a problem-solving process by searching through a state space, where the states represent the design solutions. The two most prominent techniques employed in producing "good" designs are *Simulation* and *Optimization*. **Simulation** is a problem-solving approach to design in which a design solution is postulated and tested to obtain a performance level. If the design is not found to be satisfactory, then some other solution is postulated and the process is repeated. In most cases the process ends when a seemingly satisfactory performance level is reached. **Optimization** is an approach that seeks not only a design, but the "best" design under the given criteria and constraints. Optimization may be based on a pre-defined objective function, but sometimes objective functions are very difficult to specify and do not reflect the real design intent; in such cases the optimization can also be **interactive** as in our case.

1.2 Conceptual Design

Design has been viewed as "a purposeful, constrained, decision-making, exploration and learning activity" [5]. *Decision-making* implies a set of variables, whose values have to be decided. *Exploration* involves changing the problem space within which decision-making takes place. *Learning* implies a restructuring of knowledge. It serves as a way to improve the performance of a system from one task to the next, but learning also continues throughout the design process.

Today's CAD system begins only after the concept has been fully defined. The design knowledge is ill-defined : the design space is redefined during the design process as knowledge is gained and refined (Fig.1.4). All these changes are lost to the CAD model, which can describe only the solution chosen. During design refinement or editing, the design history can be critical.

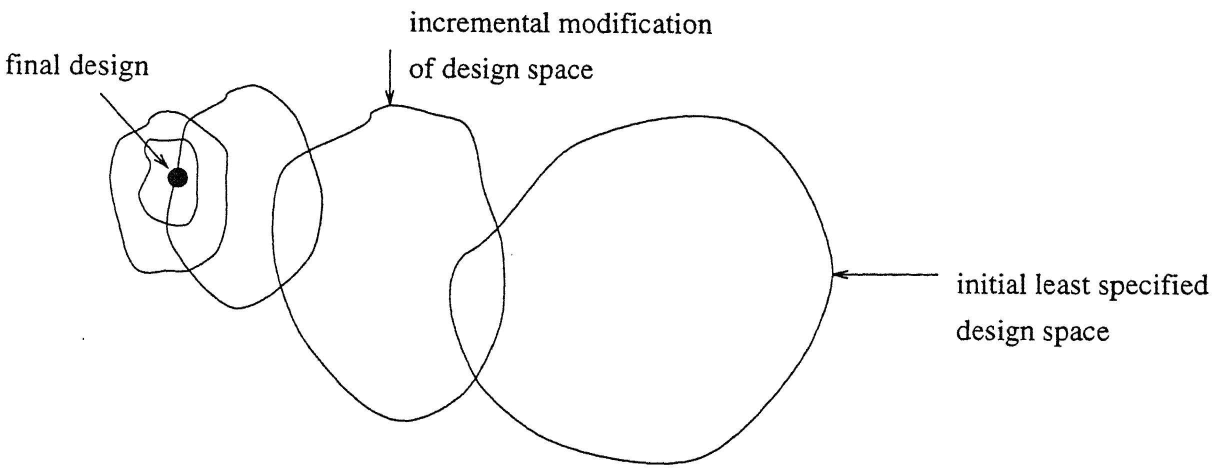


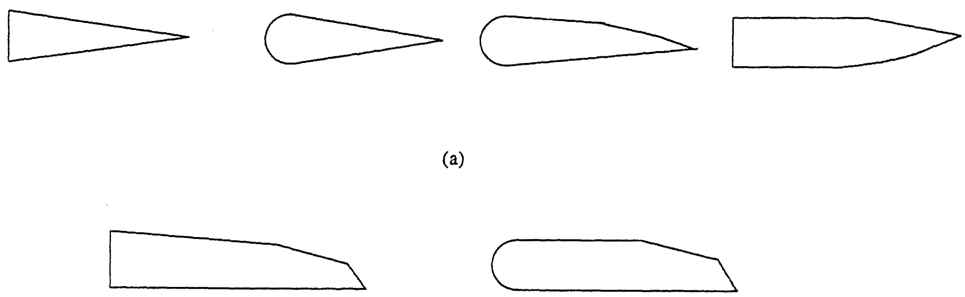
Figure 1.4: Design spaces are redefined as the designer gains more knowledge. Note that the design space is not "narrowed down" but may also go outside the previous bounds.

Hence, the definition of the design space is *dynamic*, and the design process involves not only search within such a space but also the modification of the design space itself. For CAD to reach its full potential, it must enable an incremental modification of a broad design base (space) to a single final design.

Of neccessity, designers often begin designing before all the relevant information is available. Indeed, since design is often characterized as an exploration process, what is relevant manifests itself only as the design proceeds, and changes direction even from the starting point. Thus, designers need to begin their work with only a minimal set of information, which is used to guide the introduction of more detailed information.

Since the designer begins with the minimum required information, it is not possible to define the end result outright. One of the examples that Coyne et al [5] has discussed in this context is the design of a knife. Following [5], let us look at the design of a knife blade cross-section (Fig.1.5).

The cross-section of the blade can be a simple sharp taper which defines shape



(a)

(b)

Figure 1.5: The knife blade cross-section.

class (a). Upon reconsideration, it is felt that the tip may lose sharpness too soon so that a bevel is proposed. This leads to shape class (b) with a bevel. Each of these design spaces involve many possible design instances of which the user may choose to optimize on one based on the aspect ratio or some functional attribute, but the basic shape characteristic (e.g. "bevel at tip") must remain. A design sketch similarly represents a class of blades i.e., not a single geometry but a class of similar shapes. This is typically the *abstraction* inherent in the designer's mental model. However, even at the conceptual stage, sketches generally provide more detail in the areas of the shape that are the object of attention like the "tip". Traditional optimization models fail to capture this wide variation inherent in a shape class represented by a sketch because they work with parametrizations based on a nearly finished shape. At this point, the design may already be so constrained that conceptually different designs may be excluded from the search space. Conceptual design, therefore, assumes greater importance because it provides a broader base for optimization than most parametrized models. Good designers do so much better than the best CAD not because they can optimize better, but because their conceptual design space is so much richer.

In addition to shape information, sketches also convey analogy, focus in on nov-

elty, criticality etc. Also, while we have focussed on only the geometric aspects of conceptual design, it is important to note that geometry provides only one of several aspects. Other important factors such as the knife's physical attributes (linearity of the blade, its alignment with the handle, its length and cross-section etc.), and functional attributes (sharpness, portability, configuration etc.) may have to be considered. However, geometry is probably the most difficult of these aspects to generalize.

Chapter 2

Ambiguous Shape Modeling

Consider some geometric descriptions :

1. The tolerance must be in the range 2.9 mm to 3.05 mm (Tolerance)

2. Class of all chairs (Visual recognition)

3. Shape of a tree (Graphics)

4. The stripes on a cat's tail (Mental Imagery)

All these evoke a specific geometric characterization but clearly there can exist many instantiations for objects in these classes. The associated tasks share with conceptual design, a need for representing, modeling and performing tasks with ambiguous shapes, an area that we may call **Ambiguous Shape Modeling** (ASM).

2.1 Representation schemes

In classifying representation schemes, a distinction is made between schemes which are information preserving and schemes which are information nonpreserving [7]. Information preserving representations allow for a reasonable reconstruction of the object given the shape descriptors, while information nonpreserving schemes only retain the general structure of the object and do not contain sufficient information to reconstruct the object. Most information nonpreserving schemes are qualitative and involve abstraction in an attempt to model only a smaller set of relevant features. As a consequence of

the abstraction/ information sparseness, such models are ambiguous, being capable of many valid instantiations.

Ambiguous or information nonpreserving models can be well-posed or ill-posed; e.g., tolerance models as in $2.9 < \phi < 3.05$ is an well-posed ambiguity, of which one may construct a binary valued membership function. The shape of a tree, at its extremes, is hard to define, and is an ill-posed ambiguity, as are other mental models. Conceptual design also belongs to the ill-posed category. Our tools, on the other hand, are computational and are of necessity, well-posed.Thus, we are seeking well-posed models of an ill-defined shape class embodied in a design concept.

Modeling ambiguous shapes is characterized by a trade-off between the compactness of the model (degree of abstraction) and the spread (loss of accuracy). By abstraction, we mean the ability to discard useless information in an attempt to minimize the knowledge that must be kept about an object. While the size of the model in bits can serve as a model of compactness, accuracy is much harder to quantify, especially for geometric shapes. On the other hand, using models that are excessively precise, result in characterizations like – "Anurag's height is 5 ft. 11.213271 inches" which is physically meaningless. Finding the "right" level of precision remains the fundamental problem in a wide range of problem-solving tasks.

• Continuously refinable accuracy : The "spread" in an Ambiguous shape model should be continuously variable from a wide range initially to the single final design. This also provides a design history of the constraints added from which one can branch out to generate other conceptual models based on different design decisions.

• Non-uniform granularity : Different parts of the model may have different accuracy (or minimum resolution), as in a design sketch. The ability to represent this is important to mental imagery tasks.

• Visualization : Another important aspect in ASM is that of visualization, i.e., reconstructability of the model. Since the model has many possible instances, it is difficult to display the whole model. Reconstructability makes it possible to identify representative sample instances and display these. The reconstructed instance is also

useful for FEM modeling, simulation, graphical animation, functional testing etc.

2.2 Qualitative modeling

Until recently, most object models in CAD were quantitative in nature i.e., represented by an exact boun dary. One of the alternatives to the *precise numbers* of Quantitative modeling is Qualitative modeling which proposes to use qualitative zones like sign algebra (-, 0, +), as opposed to numbers. These models represent an abstraction of reality which does not rely on exact information.

In the knife example. qualitative constructs can express aspects such as "handle is aligned with the blade" "the cutting edge is parallel to this axis" "the blade is straight" "blade is longer than the handle" (Fig.2.1(a), (b) and (c)). Similar concepts such as "blade is shorter than the handle" (Fig.2.1(d)), "cutting edge is not parallel to the axis" (Fig.2.1(e)), or "the blade is curved" (Fig.2.1(f)) etc. can also be handled. However, it is not possible to say that "the blade is not greater than 2 ft." because this is a quantitative statement, yet without it one can not rule out a sword (Fig.2.1(g)) from this class of knives.

Even though qualitative models are beneficial in a number of areas, their applicability in geometric applications has remained limited due to problems such as the inability to reconstruct the shape for visual display. For example, the well known *geon* model [3] provided an excellent abstraction for cognitive visual processes, but since geons could not be displayed – they were not "reconstructible" – the user is unable to visualize the shape being described. Clearly, the capacity for visualization is one of the most important aspects in any model. Other researchers have attempted various qualitative models to represent spatial aspects, but most of these do not attempt shape abstraction beyond models using rectangles or circles (See [11] for a review).

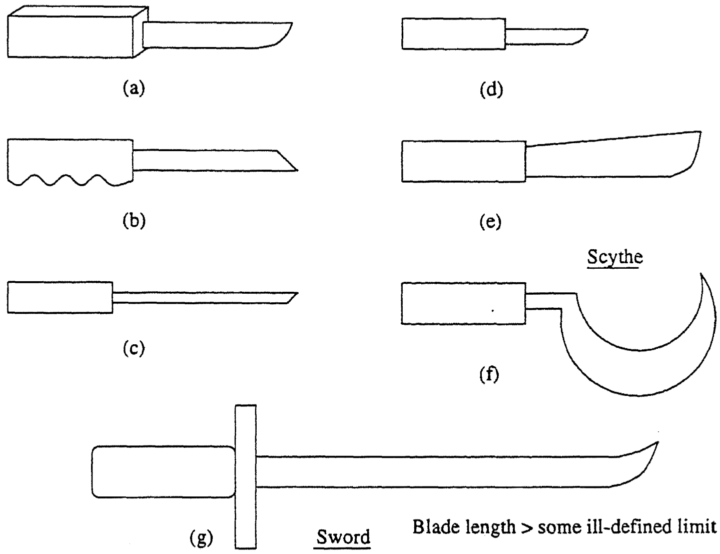


Figure 2.1: Instances from the class of knives.

2.3 Representing Ambiguous Shapes

Various representations like B-rep, CSG, sweep, grids etc. [13] exist for representing shapes. One can attempt to construct ambiguous models by allowing the parameters in these models to vary. However, such attempts to modify these models often prove to be inadequate. CSG can be ruled out since allowing the parameters to vary would result in impractically complex algorithms for Boolean operations. Grid models, already inexact, are a poor reflection of the boundary features (e.g., "bevel" in knife). B-reps have often been used for modeling ambiguity especially in parametric forms. For example, consider the model in Fig.2.2. The spatial binding in B-reps ultimately resides in the coordinates of the vertices and all shape constraints ultimately reflect on these. Sometimes, it can be quite difficult to design the constraints on the vertices. If the vertex is constrained to lie in a circular zone $\vec{X} - X_0 < d$, say, a number of undesirable effects can ensue (Fig.2.3).

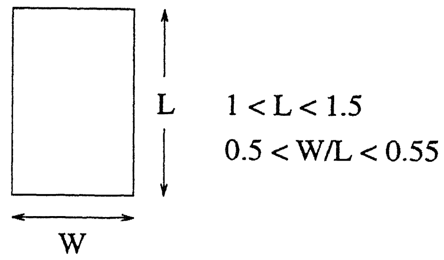


Figure 2.2: Ambiguity in B-rep models.

The sweep models are different from the boundary models, since these represent an integration process, as opposed to the differentiating process by which the boundary is produced. Differentiation results in noisy data, and integration in a *smoothing* effect. Sweep models for creating any type of shape is captured in the idea of the Medial Axis Transform or the line-site Voronoi diagram, which we discuss in the next section.

2.4 Voronoi Diagrams and Medial Axis models

Voronoi diagrams are well known tools in Geographic Modeling and other areas. "The Voronoi diagram for a set of sites $\{e_i\}$ in domain D is the set of points p, where each p is equidistant from two or more elements and where the circle centered at p and inscribing the elements does not contain or contact any other elements" [4](Fig.2.4).

If some of the sites are line segments, the V.D will in general, consist of lines and parabolas (Fig.2.4(b)). If the sites define a closed contour, then the part of the V.D inside the contour is also known as the medial axis, symmetric axis, or skeleton. The skeleton can also be said to be the locus of the center of inscribed circles that touch the boundary in at least two points, and this is the sense in which the shape is a sweep from a circle.

For polygonal contours, the MAT is a planar graph consisting of straight lines and parabolas. Some sample polygons together with their MATs are shown in Fig.2.5.

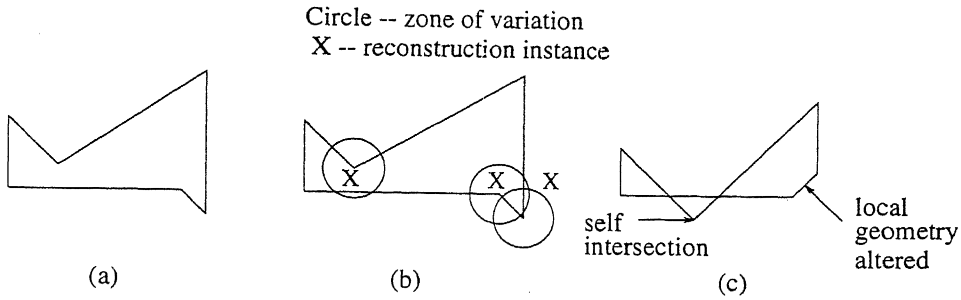


Figure 2.3: Varying the distance within the bounds leads to undesirable effects as in the reconstruction shown in (b).

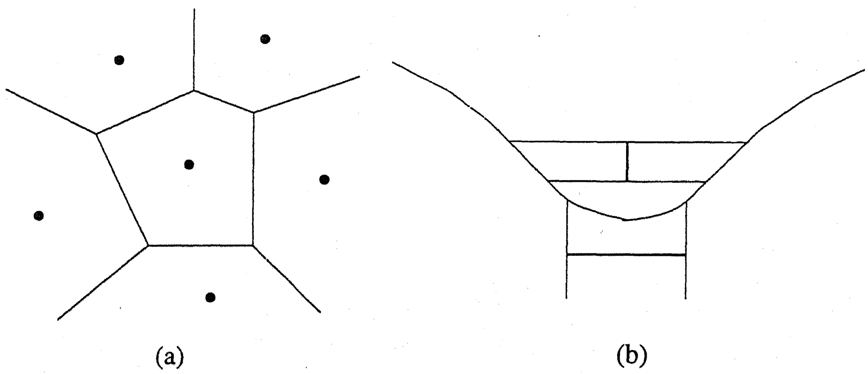


Figure 2.4: Voronoi diagram for (a) point sites (b) line sites. The thick lines in fig(b) represent line sites.

It is to be noted that mapping from polygon to MAT is a bijection (one-to-one) i.e., given a MAT together with its radius information, its polygon can be generated uniquely and vice-versa. Hence, MAT can be used to represent the shape of an object, and indeed it is a popular model in pattern recognition. However, constructing the axis from the boundary is not well-behaved since small changes in the boundary can result in structural changes in the MAT (Fig.2.6).

Obtaining the MAT from the boundary involves differentiation ; the boundary from the MAT is its inverse, integration, and far more stable.

For the purpose of treating ASMs, it is easier to perturb the axial models and still

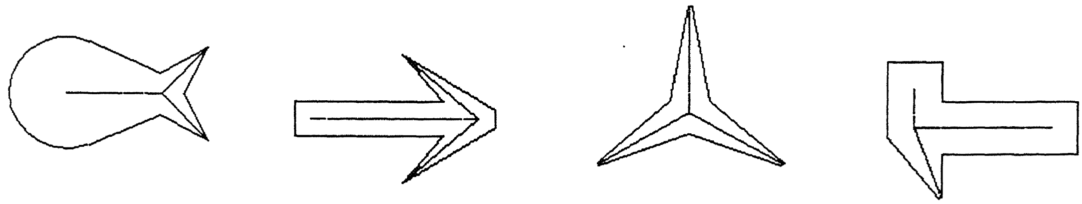


Figure 2.5: Sample polygons along with their MATs.

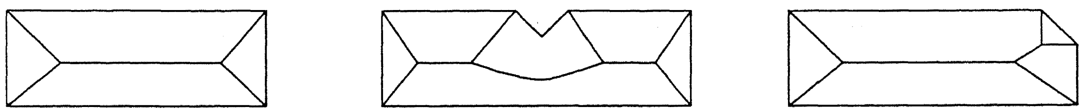


Figure 2.6: Disturbing the boundary causes structural changes in the MAT.

remain consistent. Another advantage of the MAT is that by varying the accuracy associated with the constraints, differing emphasis can be placed on different parts of the contour, which meets the non-uniform granularity criteria. Furthermore, by refining the parameter variability zones as the design progresses, one can achieve refinable accuracies.

The question of reconstructing a shape instance from a perturbed axis model has been attempted in [6]. The model is capable of abstracting the shapes to many levels of abstraction and has been used in shape similarity tests on human subjects. However, this work was limited to convex shapes only due to the difficulty of constraint satisfaction. Recently, a hybrid qualitative medial axis transform (HQMAT) approach with reconstruction and visualization capabilities for handling general shapes including non-convex ones has been developed [2]. This approach also uses more general constraint satisfiers such as Genetic Algorithms (GA). This model is the base for our work

and is discussed next.

2.5 Hybrid Qualitative Medial Axis Transform

Following Agarwal et al [2], the structure of the MAT has been simplified for introducing a discretization and variability in its parameters. An MAT can be represented by a list of straight line link segments which are segments along the line-site-Voronoi-diagram. The data required for constructing a shape from its MAT consists of :

1. link lengths,
2. link orientations,
3. node radii, the radius of the maximal circle inscribed at the node.

In HQMAT, each of these is represented not by an exact value but a zone of variation. This has been reflected in Fig.3.3. To generate different shapes belonging to the same class one can change link lengths, the angle between links, or the radii associated with nodes. Here, the MAT is used as the basic model on which to construct a model for a qualitative shape which embodies the shapes to be queried for. The qualitative model then relates all the link lengths and angles using a 2-D spatial reasoning model [9]. Unlike a purely qualitative model, the variability in the parameters are bounded for meaningful results by defining a suitable discretization (on a quantitative space). The discretization used, and the computations on it, is based on the hybrid qualitative algebra [8], which refers to a discretization that is obtained by subdividing the qualitative zones. This provides variable resolution at different parts of the contour. Hence, with the Hybrid version of the MAT, the lengths of each link and their radii are stored as scaled ratios of a nominal length.

2.6 Consistency

The problem with having inexact ranges for the MAT parameters is that some sets of parameter instantiations may result in inconsistent data from which no contour can

be generated. An MAT is *infeasible* if no boundary can be constructed for these axis specifications [1] (Fig.2.7). An MAT is said to be *inconsistent* if a boundary for the MAT can be generated but the MAT of this boundary is different from input MAT (Fig.2.8).

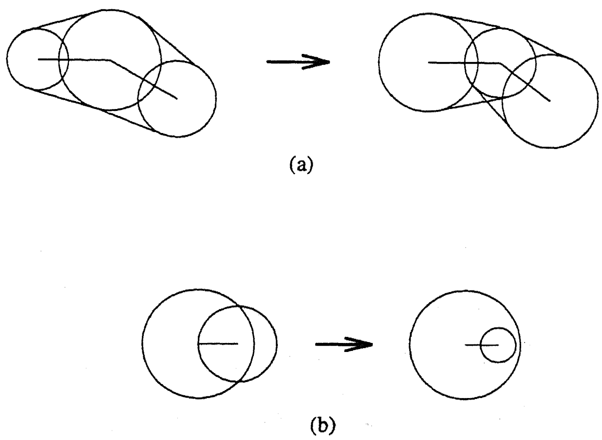
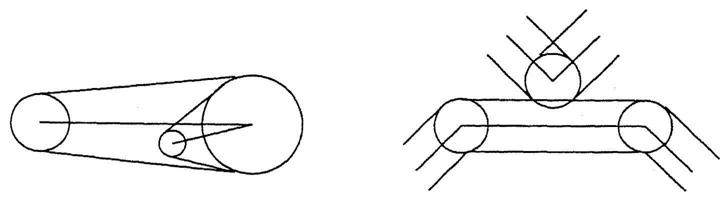


Figure 2.7: Infeasible MATs. (a) end radii becoming too high. A part of the boundary is included in the circle (b) Full inclusion (radius too small).

To eliminate such inconsistent/ infeasible shapes, penalty functions coupled with Genetic Algorithms (discussed in the next section) have been used. Each individual in a GA population represents an instance of shape generated by the perturbed MAT model. Thereafter, it is evaluated in terms of its fitness criteria.

2.7 Genetic Algorithms (GA)

Genetic Algorithms are well known as a mechanism for searching non-monotically in non-linear domains. Following Mukerjee et al [12], we can describe GA as taking a population of individuals from the search space, evaluating these for their effectiveness in meeting the constraints, and then generating a new set of individuals by cross-linking some of the traits in the more successful individuals of the previous generation. This



(a) (b)

Figure 2.8: Inconsistent MATs. (a) link inside link (b) arc intersects contour.

technique has been shown to be a surprisingly robust form of nonlinear search. The "gene" in this process is a binary or real coding of the search parameters, and the "fitness function" is the degree to which the individual succeeds in the task. One of the advantages of the GA process is that the search is independent of the structure of this fitness function – it can be continuous or discrete, and may even be interactively determined by the user, or obtained from real or simulated runs on the current data point.

Since GAs use a population of solution points distributed over the search space, it is easy to provide a display of many shapes simultaneously. However, like all non-linear search algorithms, a certain amount of tuning is needed in parameters such as population size, crossover, and mutation.

2.8 Design Refinement

Design refinement is associated with selecting a design in the design space after taking into consideration additional factors such as the physical and functional attributes of the desired shape.

2.8.1 Pre-defined Optimization

Traditional Optimization

Traditional design optimization, as mentioned earlier, works with parametrizations based on a nearly finished geometry. The design decisions (assumptions) by which this quasi-finished shape was arrived at is often outside the purview of optimization processes. This restricts the solution space available to the optimization tool, and consequently, the quality of possible enhancements. Standard optimization techniques are slow in constraint satisfaction and standard constraint satisfaction techniques are inefficient in optimization.

Here, by using a multi-individual, non-linear optimization tool (GA), a parametrization with a greater degree of geometric variability than can be captured by traditional optimization methods has been provided. Thus the solution space is broader and better optimization may be possible. GAs have the advantage that they can be used for 'constraint satisfaction' and 'optimization' both i.e., as a search and optimization tool.

GA approach

An inexact or ambiguous model has its "range of ambigüity" defined by a system of constraints [12]. In this case, many of these constraints are inherent from the discretization adopted, but some of them are in the form of holistic attributes of the desired shape. Traditional constraint solvers attempt to provide a single solution, but this model [2] provides a large number of "feasible" shapes to display the space of shapes rather than a single one.

Constraints may be assigned priorities or weights and may interact in unpredictable ways. Further, the interaction between the constraints is so nonlinear that small changes in the input set may lead to large changes in the results. Some constraints are interactive, and others are discrete but necessarily pre-defined. In order to display a diverse batch of reconstructions (a population), the GA approach has been preferred.

In addition to the basic shape, the user may have many other aspects that he would like to control. The contour may be sharp, or it may be smooth. It may be symmetric

or not. It may or may not be convex. All these factors can be incorporated into the fitness criterion for the GA. The objective then becomes to select shapes that are closest to the user concept. The following properties of the contour have been taken into account in this model for its fitness evaluation (following [1]),

1. Area of the polygon
2. Smoothness
3. Penetration of generating circles
4. Non-convexity
5. Number of edges
6. Aspect Ratio
7. Parallelism/ Perpendicularity of edges

For each individual, a reconstruction of the outer contour is obtained. Each fitness criteria is modelled as a constraint on this contour itself. The user can adjust the relative importance for each of these properties, and can also define new contour based constraints. Every random MAT generated by GA is evaluated and given a suitable penalty value according to its consistency and the user-defined criteria. Use of GA with the objective of minimizing total penalty leads to shapes which are most desirable.

2.8.2 Interactive Optimization

In addition to the constraints mentioned in the previous section, we would also like to have constraints that are not expressible as equations – just a "Ah – this is just what I want", or "I do not want this". Also, after a few sample generations, the user may want to branch out in a particular direction by redefining the range for some parameters about which he has made up his mind at this point. That is, the user may want to redefine the design space by changing some parameters in order to propagate

in a different direction. These are some of the requirements that we address in our current work. In effect, we have tried to provide a base for interactive optimization.

Chapter 3

Results

3.1 Implementation

We present some empirical results in this chapter based on the SHAPE_OPTIM program. This has been implemented on the X-Windows platform on HP 9000 systems. In this work, we have limited ourselves to three-link MAT cases, the longest of which is taken as the reference length, and the length ratios of the other two are l_2 and l_3. Similarly, the radius ratios at the four nodes are r_1 through r_4. The two angles made by l_2 and l_3 with the reference link are θ_1 and θ_2 (Fig.3.1).

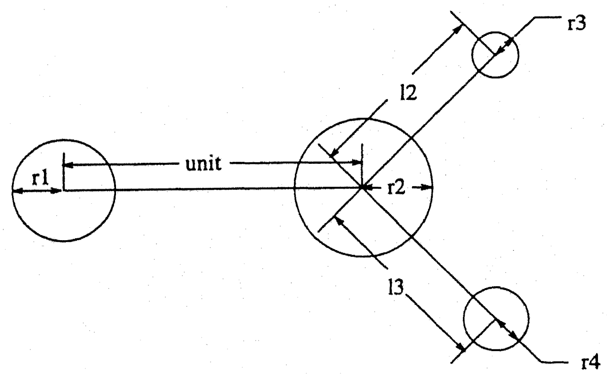


Figure 3.1: The basic geometry of a three-link MAT.

Agarwal's work [1] provided a mechanism to determine desirable shapes from the class of shapes represented by QMAT using GA and to remove the infeasible or inconsistent shapes. However, the optimization criteria had to be pre-defined and the process had to be restarted each time a criterion were changed.

The principal contribution of this work is in providing an interactive mechanism so that a designer can try to come closer to his ill-defined concept space by viewing the contours interactively. We have provided an interface for the shapes to be displayed *dynamically* to the user and for interactive optimization. This allows modification of the design space and the designer can interactively control the propagation of the design changes. Also, as mentioned earlier, refinable accuracies can now be achieved by refining the parameter variability zones as the design progresses.

Fig.3.2 shows the process flow for MAT based ambiguous shape modeling. In this work we have started directly with HQMAT input, i.e., the regions of variability defined by the initial parameter ranges. This is displayed to the user (a screen dump is shown in Fig.3.3). This is followed by the individuals generated in each generation. In practice, the user may *sketch* a sample shape, see its MAT, and then apply the ranges of variability to generate the initial data. For any given generation, SHAPE_OPTIM enables the user to select any of the figures and enlarge (zoom) it (for better viewing), view its contour or MAT, or see the best figure within the defined constraints straightaway. The user can display the same generation repeatedly before proceeding to the next one. When the run is on, the user may feel the need to retain some particular traits in the successive generations. The interface enables him to emphasize these traits interactively, or if he is a more sophisticated user, then by editing the relations defined earlier by him. The user can save any figure and quit from the program at any point in the procedure.

3.2 Interactive and Non-Interactive mode

In the **non-interactive mode**, the user specifies the holistic shape criteria and their weights to be used in generating the shapes. This defines the optimization criteria

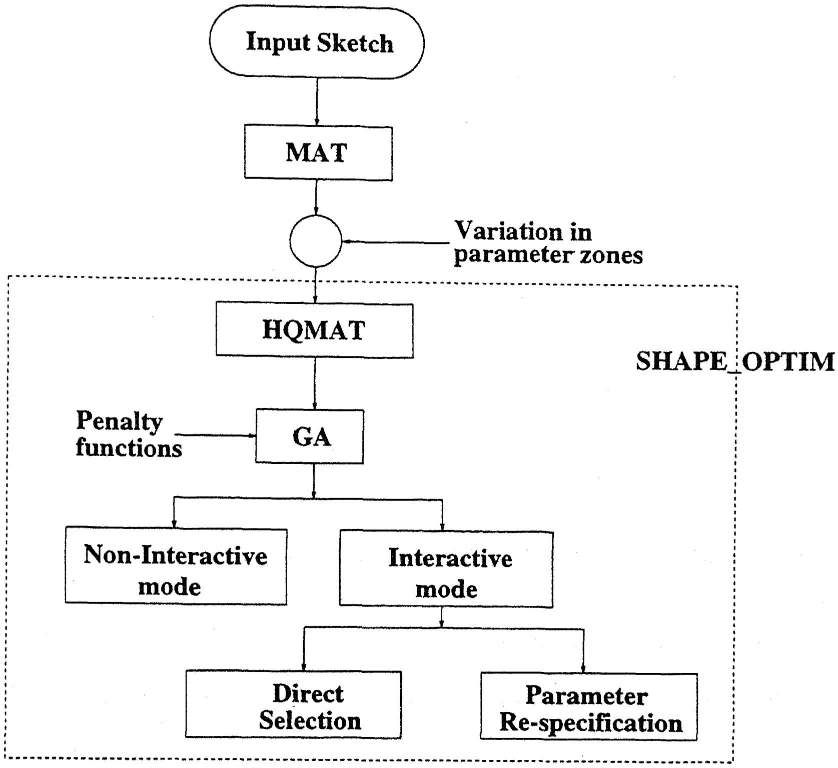


Figure 3.2: Overview of the flow process for MAT based ambiguous shape modeling. The dotted line shows the domain for the program SHAPE_OPTIM.

under which the shape is modified as a pre-defined optimization.

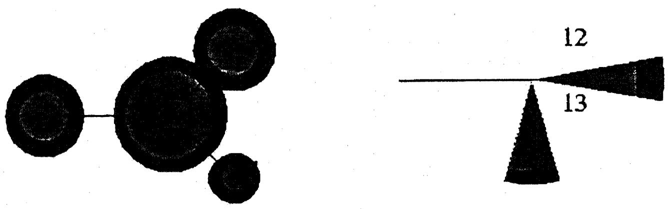
In the **interactive mode**, the user is shown generation after generation of the shapes being generated. At any point in the procedure, the user can emphasize certain characteristics in a particular shape displayed by clicking on it, which reduces the penalty associated with the figure, thereby increasing the chances that its chromosomes will be propagated in the following generations. The individuals produced in the next generation typically tend to be more towards the shape characteristics chosen by the user (section 3.4.1).

To illustrate both these procedures, we trace the development of a **rocker arm.** In each of these instances, a set of QMAT data is provided by the user (Table 3.1).

Variable	Rocker arm
r_1	(0.2, 0.32)
r_2	(0.35, 0.45)
r_3	(0.2, 0.32)
r_4	(0.1, 0.2)
l_2	(0.8, 1.0)
l_3	(0.7, 0.8)
θ_1	(-10, 10)
θ_2	(75, 105)

Table 3.1: QMAT data used for rocker arm shape.

The numbers in parenthesis represent the range of values provided by the user. The regions of variability associated with each variable defined by these parameter ranges are shown in Fig.3.3.



(a) (b)

Figure 3.3: Rocker arm : Regions of variability for each variable are shown in black. (a) shows the radius variations, and (b) the zones for the endpoint of each link. This figure and the associated table is displayed to the user at the very beginning and after any parameter edits.

3.3 Results : Non-Interactive mode

Fig.3.4 (a) through (e) illustrate several generations obtained in the **non-interactive mode**. Here, the parameter range is *pre-defined* and the user is shown the individuals

generated in successive generations. As the algorithm progresses, individuals with better fitness (lower penalty values) are retained and there is less diversity. However, to change any of the characteristics (parameter ranges) associated with these individuals, the user has to change the original input and start again.

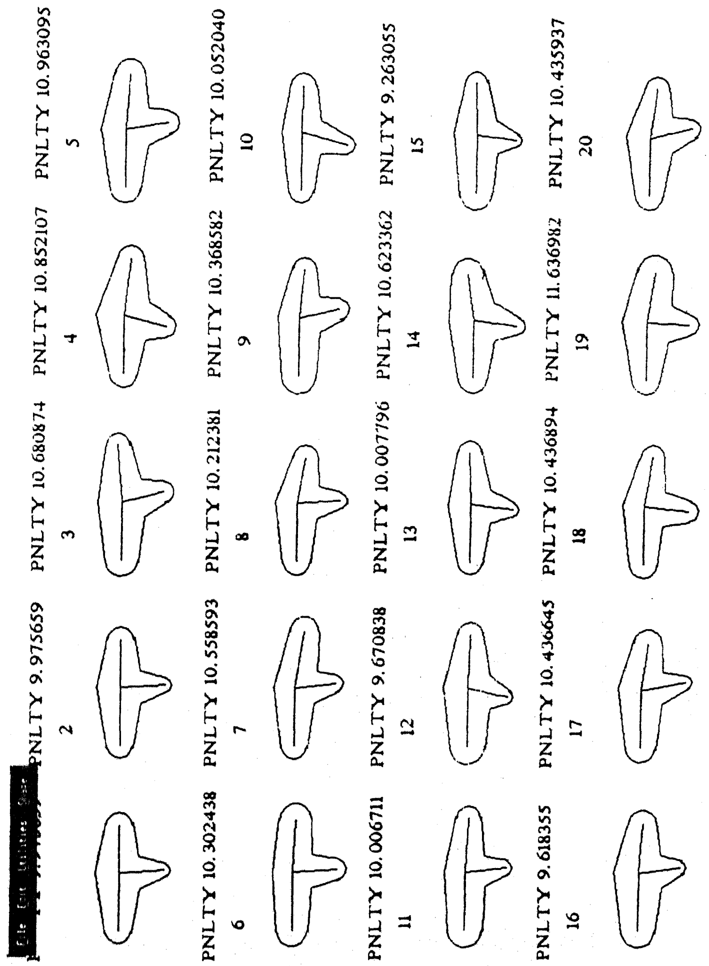


Figure 3.4: (a) Rocker arm: *Non-Interactive mode.* Generation 0. Each individual is formed by randomly choosing variables from within the limits shown in Table 3.1; there has been no optimization. Penalty values associated with each individual are shown. Average penalty 10.29. The worst, sample 19 (penalty 11.64) has poor symmetry, high area and low smoothness, while the best, sample 15 (penalty 9.26), is somewhat more along the desired criteria. Traits from the shapes with a lower penalty have a higher chance of being propagated to the next generation.

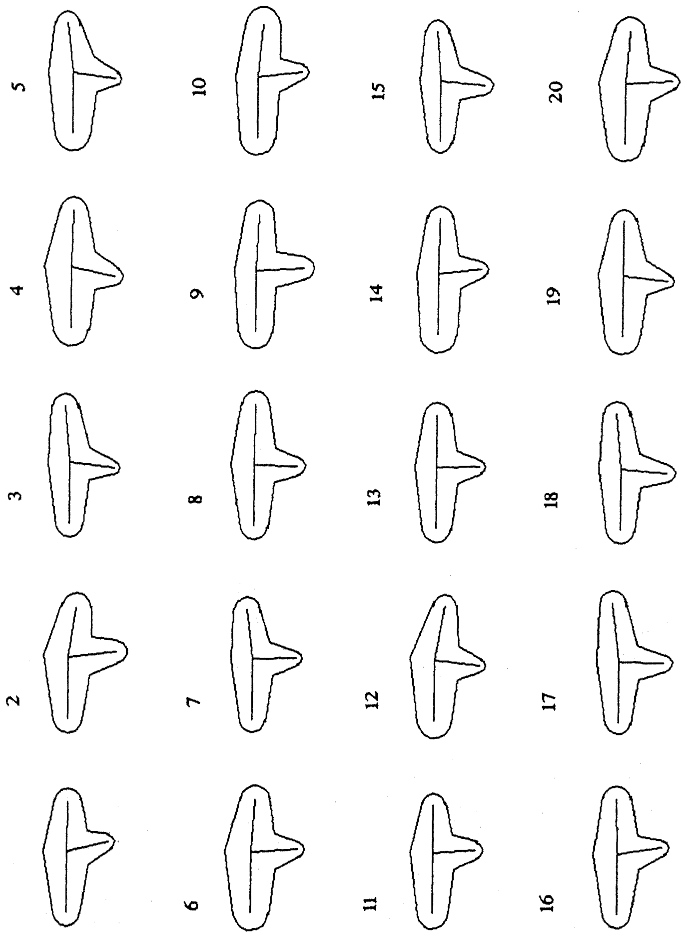


Figure 3.4: (b) Rocker arm: *Non-Interactive mode.* Generation 2. Penalty ranges from 9.0 to 10.92 (Average 10.08). Best sample is 5 (r_1=0.29, r_2=0.41, r_3=0.20, r_4=0.1, l_2=0.82, l_3=0.71, θ_1=6.89, θ_2=95.25).

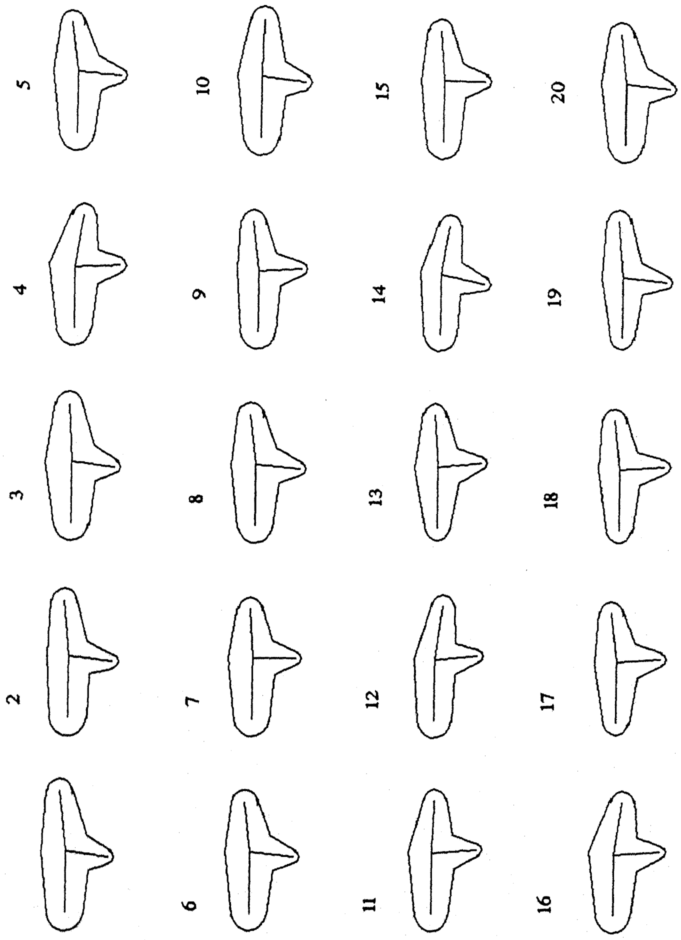


Figure 3.4: (c) Rocker arm: *Non-Interactive mode.* Generation 5. Average penalty=9.51. Maximum=10.1. Minimum=9.04. The best penalty in an earlier generation may be less than the minimum in this generation, but the average penalty typically falls in successive generations.

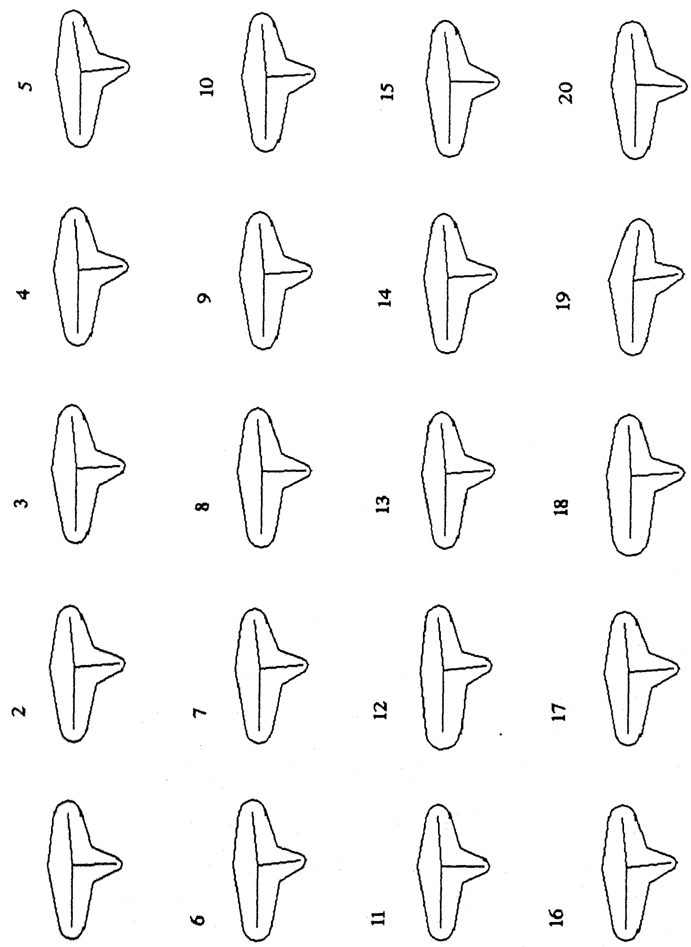


Figure 3.4: (d) Rocker arm: *Non-Interactive mode.* Generation 10. Average penalty=8.79. Penalty ranges from 8.53 to 9.25. The improved penalty reflects the better symmetry and smoothness in most of the contours. Best is sample 7 (r_1=0.21, r_2=0.4, r_3=0.2, r_4=0.1, l_2=0.81, l_3=0.71, θ_1=7.69, θ_2=84.67).

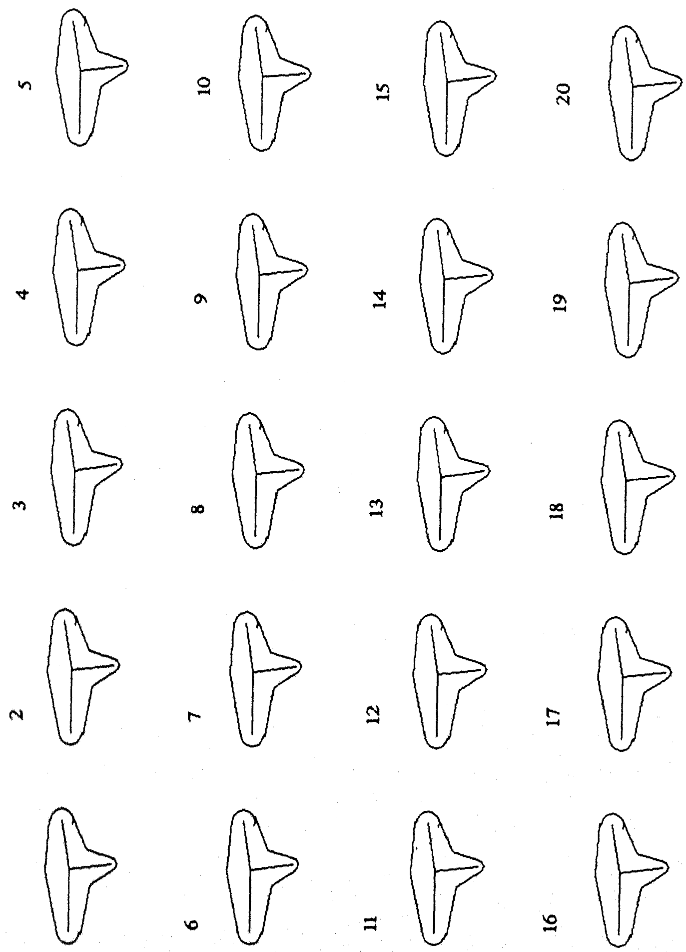


Figure 3.4: (e) Rocker arm: *Non-Interactive mode.* Generation 20. Maximum penalty=8.45, Minimum=8.43, Average=8.44. Substantial decrease in penalty range reflects the near convergence on a set of acceptable shapes. The population as a whole is less diverse than in earlier generations.

3.4 Results : Interactive mode

There are two different modalities in the **interactive mode** :

1. *Direct Selection* : Here the user directly clicks on a few shapes which appeal to his ill-defined mental concept ("Ah – just what I had in mind").The characteristics of these shapes will then be propagated in future generations. Internally, the penalties for these shapes are reduced by 10% each time the user clicks on it. A triple click will therefore reduce the penalty by a factor of 0.73, thereby making it very likely that its characteristics will be propagated in the gene pool. Fig.3.5 shows the progress of GA with each generation for the non-interactive and interactive (Direct selection) modes.

2. *Parameter Range Re-definition* : An user more familiar with the HQMAT formalism may directly choose to define the variability ranges for the various MAT parameters. This can be quicker, but very large redefinitions may result in problems.

3.4.1 Results : Direct Selection

We consider the example of rocker arm shape again. In Fig.3.6(a) is the nominal, symmetric shape. Fig.3.6(b) shows link 3 at an angle to the reference link and reflects a trait that we term as the "left inclination" of link 3. Fig.3.6(c) shows link 2 with a "downward inclination" trait. These traits are used on the basis of visual inspection and are subjective in nature.

Fig.3.7 (a) through (d) illusrate the generations obtained in the **direct selection mode**. The user is shown the first generation (Fig.3.7(a)) of the shapes generated and he chooses the figures that reflect the "left inclination" of link 3. Fig.3.7 (b) shows the same procedure being followed. At some point here, say, the user desires to retain the "downward inclination of link 2" characteristic. Hence, the individuals with this attribute are chosen (Fig.3.6 (c)). Fig.3.6 (d) illustrates a generation in which most of

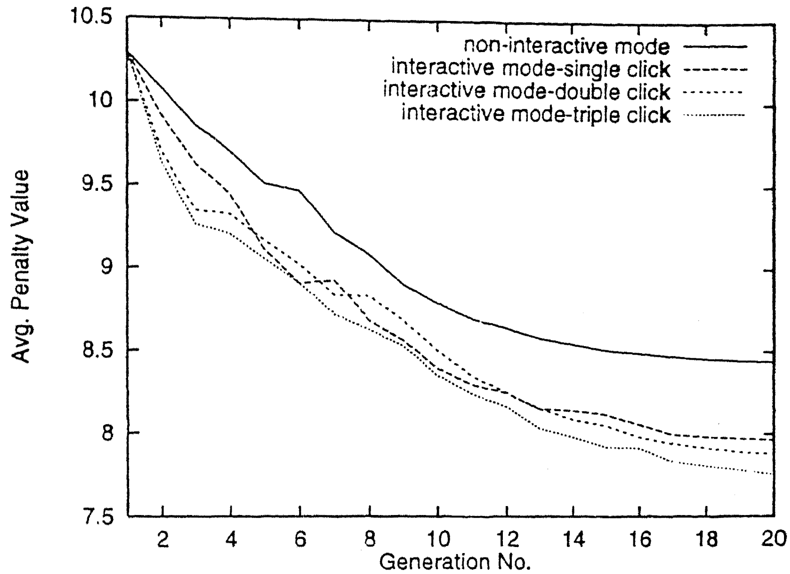
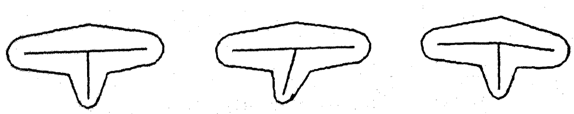


Figure 3.5: Progress of GA with each generation. Average penalty value (Generation 1) =10.29. Average penalty value (Generation 20) for (i) Non-Interactive mode=8.44, (ii) Interactive (Direct selection) mode (a) single click=7.96, (b) double click=7.88, (c) triple click=7.75. These results are only indicative since the shape populations are different in these runs and the user subjectively clicks on a few minimum penalty shapes. Since the GA uses tournament selection, the reduction of the penalty factor from 0.9 to 0.81 to 0.729 does not have a very great impact.

the individuals have the characteristics of "link 3 being left inclined" and "link 2 being inclined downward" as desired by the user.



(a) (b) (c)

Figure 3.6: Rocker arm shape. (a) Link 2 is aligned with the reference link and link 3 perpendicular to it, (b) *left inclination* of link 3, (c) *downward inclination* of link 2.

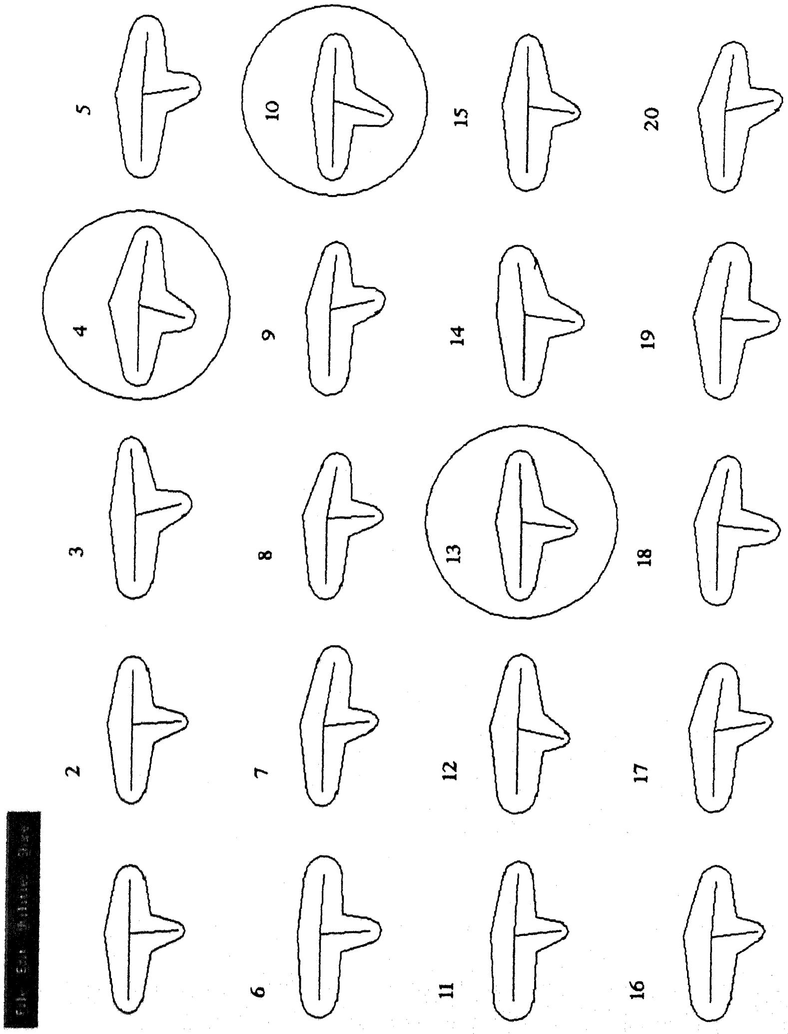


Figure 3.7: (a) Rocker arm: *Interactive mode - Direct selection.* Generation 0. The population formed randomly, from within the domain of Fig.3.3, is the same as in Generation 0 for the non-interactive case (Fig.3.4(a)). The user tries to emphasize "left inclination" of link 3 by clicking on the encircled figures. Link 3 is left inclined in 7 individuals, right inclined in 5, and neutral in 1 (subjective estimates).

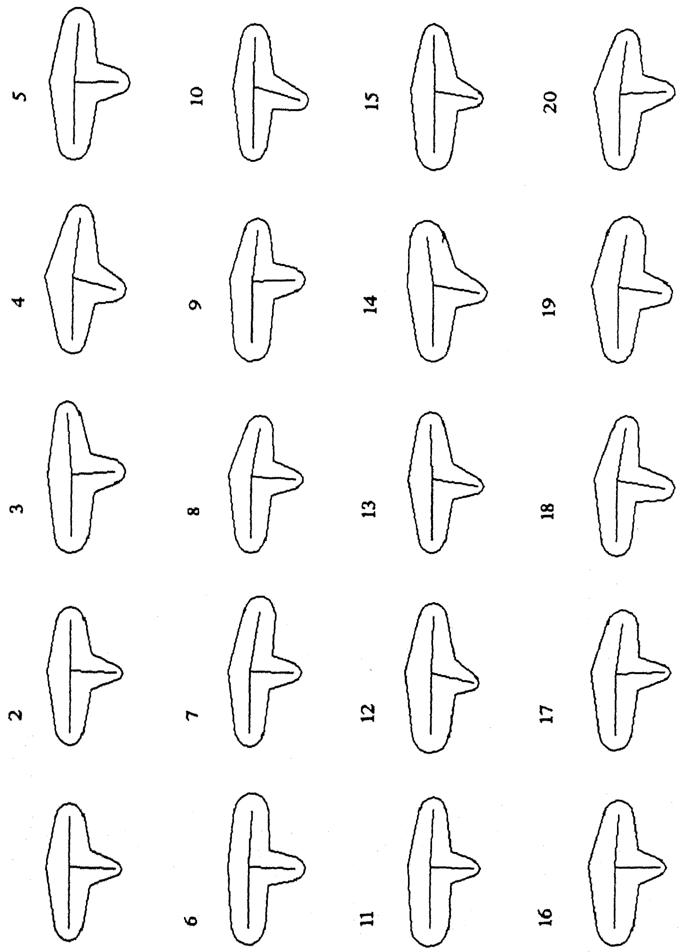


Figure 3.7: (b) Rocker arm: *Interactive mode - Direct selection.* Generation 3. The population as a whole is already reflecting a reflects a bend towards left inclination. Link 3 is left inclined in 14 individuals, right inclined in 5, and neutral in 1.

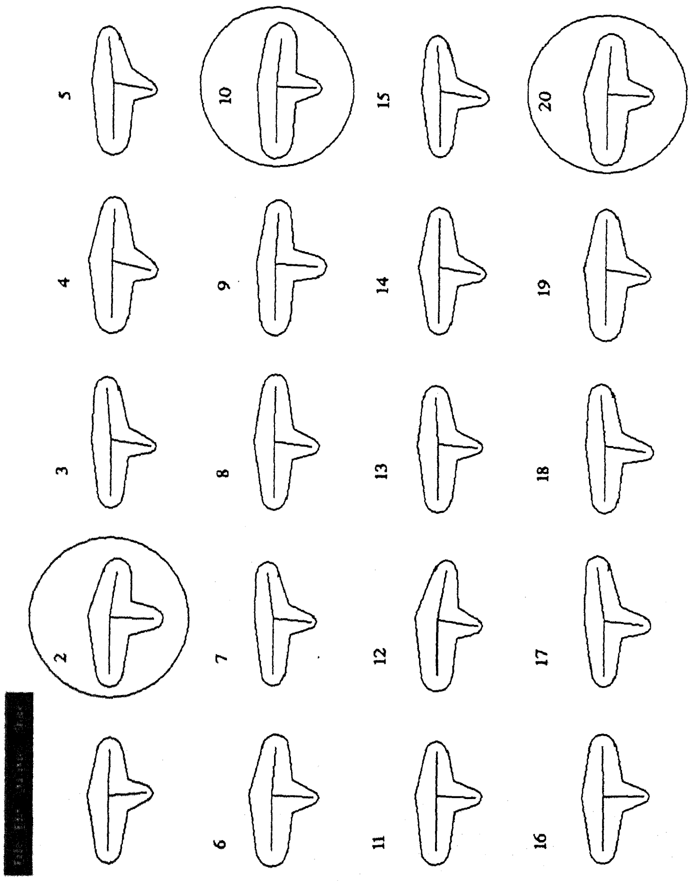


Figure 3.7: (c) Rocker arm: *Interactive mode - Direct selection.* Generation 5. Link 3 is left inclined in 16 individuals, right inclined in 1, and neutral in 3. The user chooses to emphasize "downward inclination" of link 2 by clicking on the encircled figures. Link 2 is inclined downwards in 7 individuals, upward in 7, and neutral in 6.

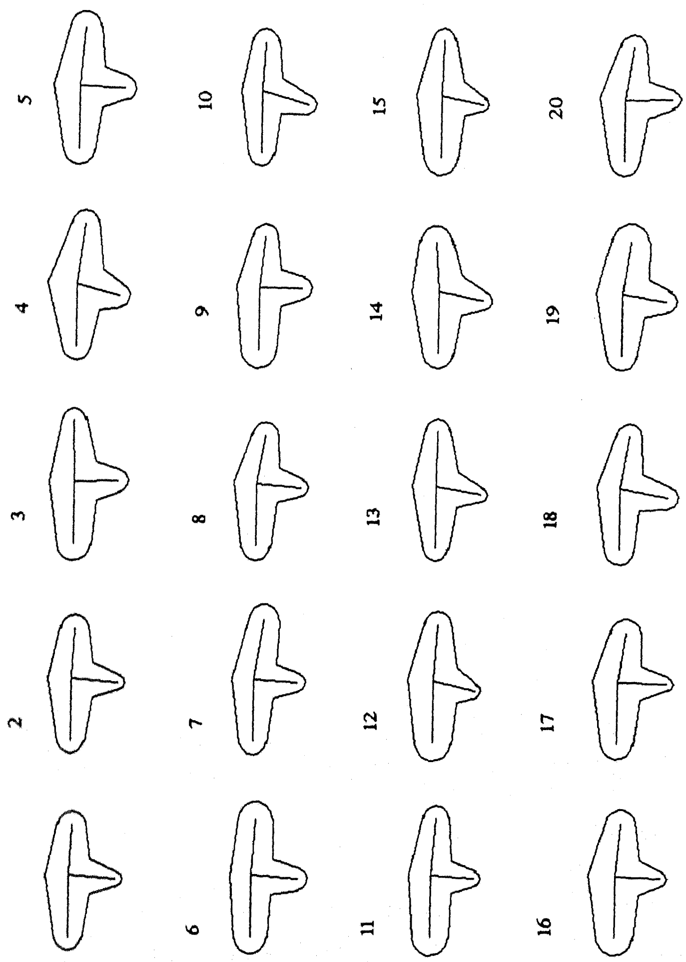


Figure 3.7: (d) Rocker arm: *Interactive mode - Direct selection.* Generation 10. Link 2 is inclined downwards in 17 samples, upward in 1, and neutral in 2. The property of "left inclination" of link 3 chosen earlier has been preserved. Therefore, most individuals show both the desired traits. 75% of the samples have link 3 left inclined and 85% have link 2 downward inclined and 65% have both the traits.

3.4.2 Results : Parameter Range Re-definition

Here, the user can edit the HQMAT parameter ranges directly. Consider the rocker arm shape example discussed earlier to retain the "left" and "downward" inclinations of link 3 and link 2 respectively. Figs.3.8 (a) through (c) show how this can be achieved by changing the parameter values.

Hence, it can be seen that by enabling the user to redefine the parameters during the process itself and without restarting all over again, the desired traits can be incorporated in the individuals in **fewer** generations than by choosing samples with those traits in each generation. This method is discussed with more examples in the next section.

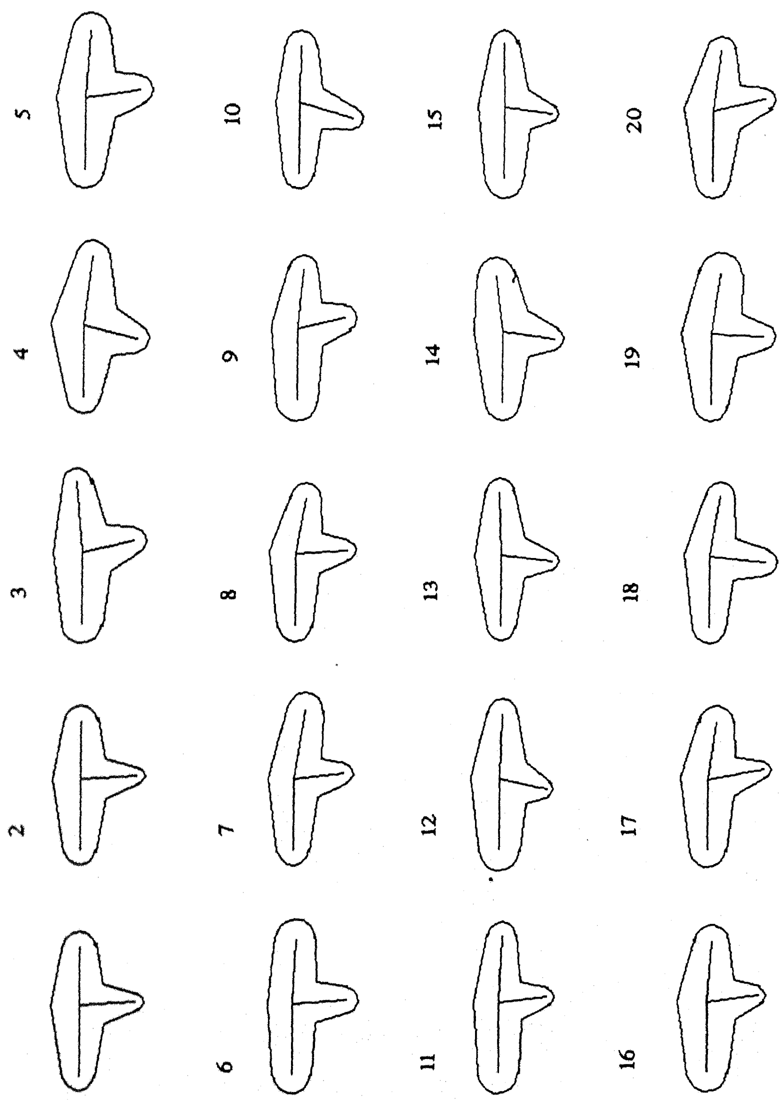


Figure 3.8: (a) Rocker arm: *Parameter range re-definition.* Generation 0. Same as earlier in Fig.3.4(a). After viewing this generation, to emphasize "left inclination" of link 3, range of values is redefined for θ_2 to be (85, 105) instead of (75, 105) and for "downward inclination" of link 2, range for θ_1 is modified to (-10, 5) instead of (-10, 10).

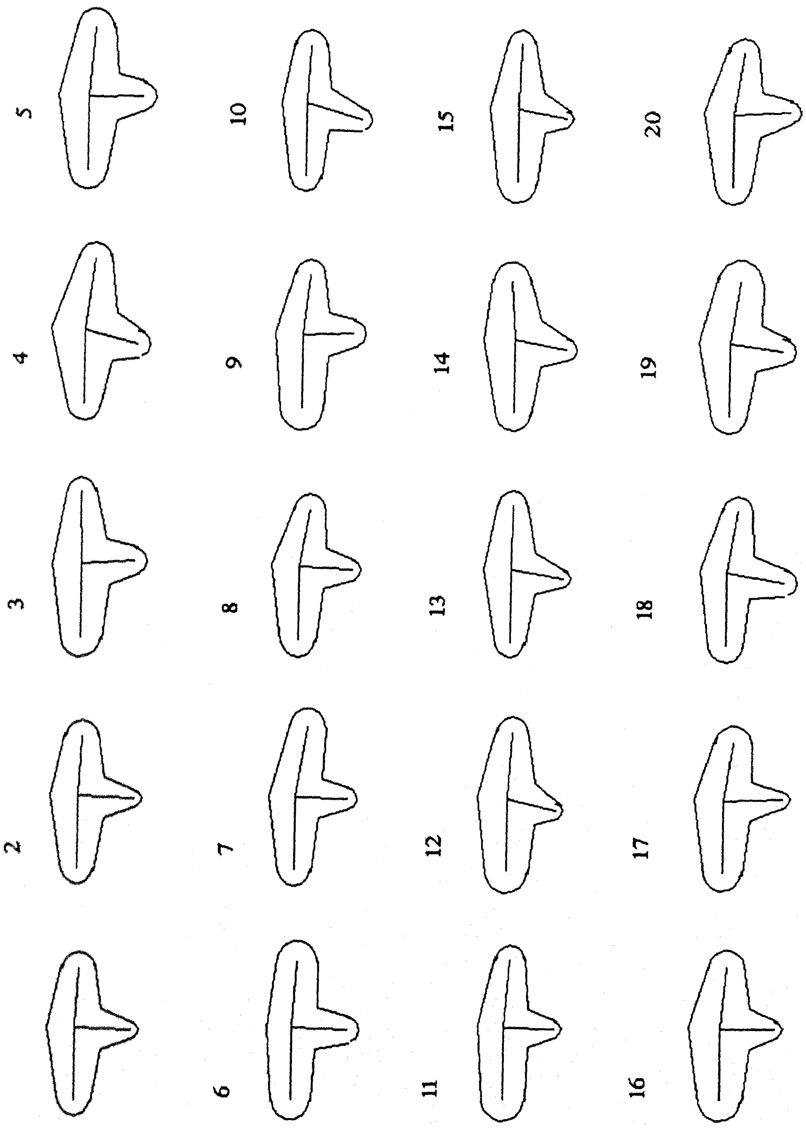


Figure 3.8: (b) Rocker arm: *Parameter range re-definition.* Generation 1. Most of the individuals show a bend towards the emphasized chracteristics. To further emphasize the "left inclination" of link 3, range of values is redefined for θ_2 to be (90, 105) and for "downward inclination" of link 2, range for θ_1 is modified to (-10, 0).

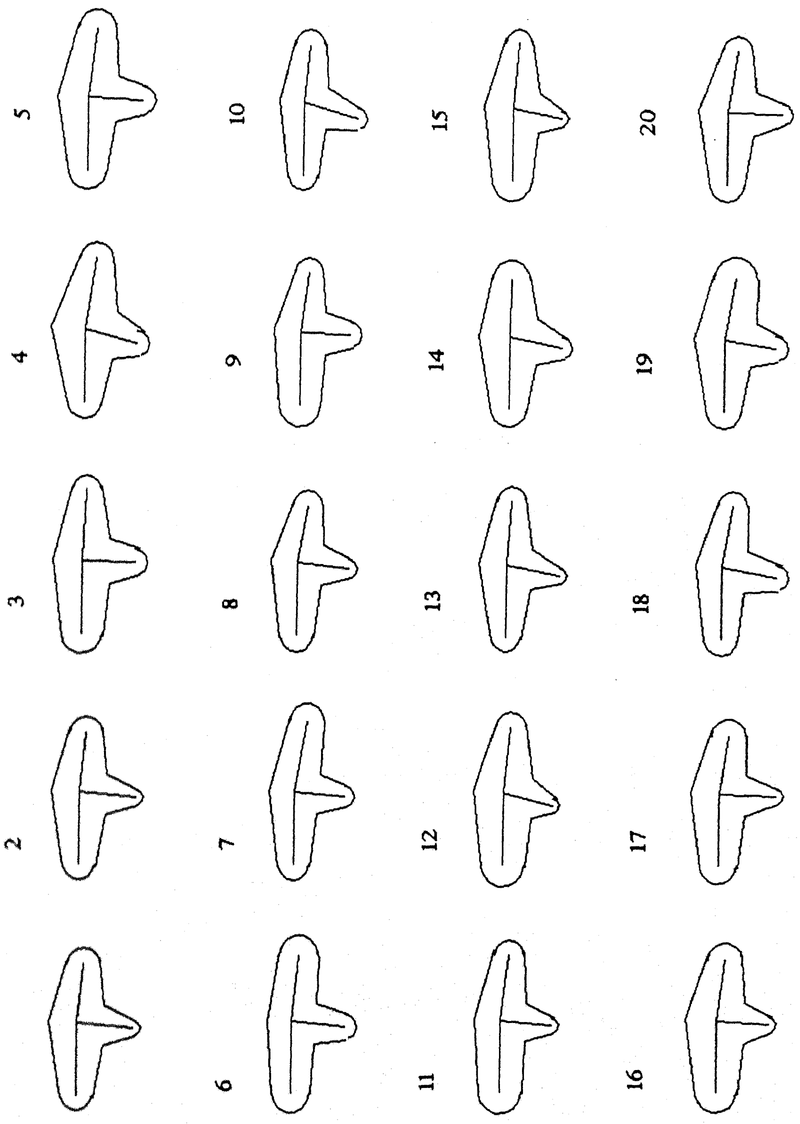


Figure 3.8: (c) Rocker arm: *Parameter range re-definition.* Generation 2. All individuals reflect a left inclination of link 3 and a downward inclination of link 2.

3.5 Re-defining the design space

The parameter re-definition process may also be used to substantially redefine the design space as the user, in the process of seeing (and possibly simulating or testing) the generated shapes, modifies the design specifications, causing him/ her to redefine the design space itself, as in Fig.1.4. Also, it enables the user to fine tune the range of parameters from a wide range initially to those representing a single final design.

Design space re-definition is illustrated with the following examples :

1. Knife to chisel : Here, the user realizes that substantial force will be applied and opts for a thicker, less angled cross-section.

2. Pistol to hammer shape : A graphic designer flits from one icon to another.

3. Mercedes logo to Tata logo : This is another graphic change.

In each of these cases, the user provides a set of data initially to define the initial shape class. Thereafter, over the next few generations, the parameter ranges are changed gradually to generate conceptually different shapes.

3.5.1 Knife to Chisel

The design of a knife is characterized by the 'bevel' at the tip as discussed earlier (section 1.2), whereas a chisel is characterized by a taper at both the edges culminating in a pointed tip. The length of the tapered edges may vary depending on the functional attributes. Fig.3.9 shows examples of knife and chisel shapes.

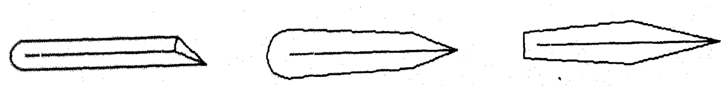


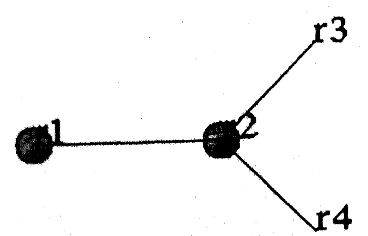
Figure 3.9: Knife and chisel shapes.

The user provides an initial shape class for the knife shape. Thereafter, changes in the parameter ranges are carried out to lead to the final shape. Fig.3.11 (a) through

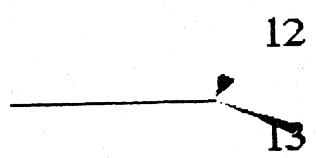
Variable	Gen 0	Gen 4	Gen 5	Gen 7
r_1	(0.08, 0.1)	(0.1, 0.1)	(0.15, 0.2)	(0.2, 0.2)
r_2	(0.08, 0.1)	(0.08, 0.1)	(0.08, 0.15)	(0.05, 0.15)
r_3	(.001, .001)	(0.0, 0.0)	(0.0, 0.0)	(0.0, 0.0)
r_4	(.001, .001)	(0.0, 0.0)	(0.0, 0.0)	(0.0, 0.0)
l_2	(0.07, 0.13)	(0.08, 0.1)	(0.2, 0.3)	(0.3, 0.3)
l_3	(0.23, 0.4)	(0.18, 0.25)	(0.2, 0.3)	(0.3, 0.3)
θ_1	(45, 80)	(60, 80)	(10, 30)	(0, 0)
θ_2	(18, 25)	(8, 23)	(-5, 0)	(0, 0)

Table 3.2: QMAT data used for knife to chisel transformation.

(d) shows the transformation from the initial knife shape to the chisel shape. The set of data for Fig.3.11 (a) through (d) are shown in Table 3.2. The regions of variability associated with each variable defined by the initial parameter ranges are shown in Fig.3.10.



(a)



(b)

Figure 3.10: Knife shape : Regions of variability for each variable are shown in black. (a) shows the radius variations, and (b) the zones for the endpoint of each link.

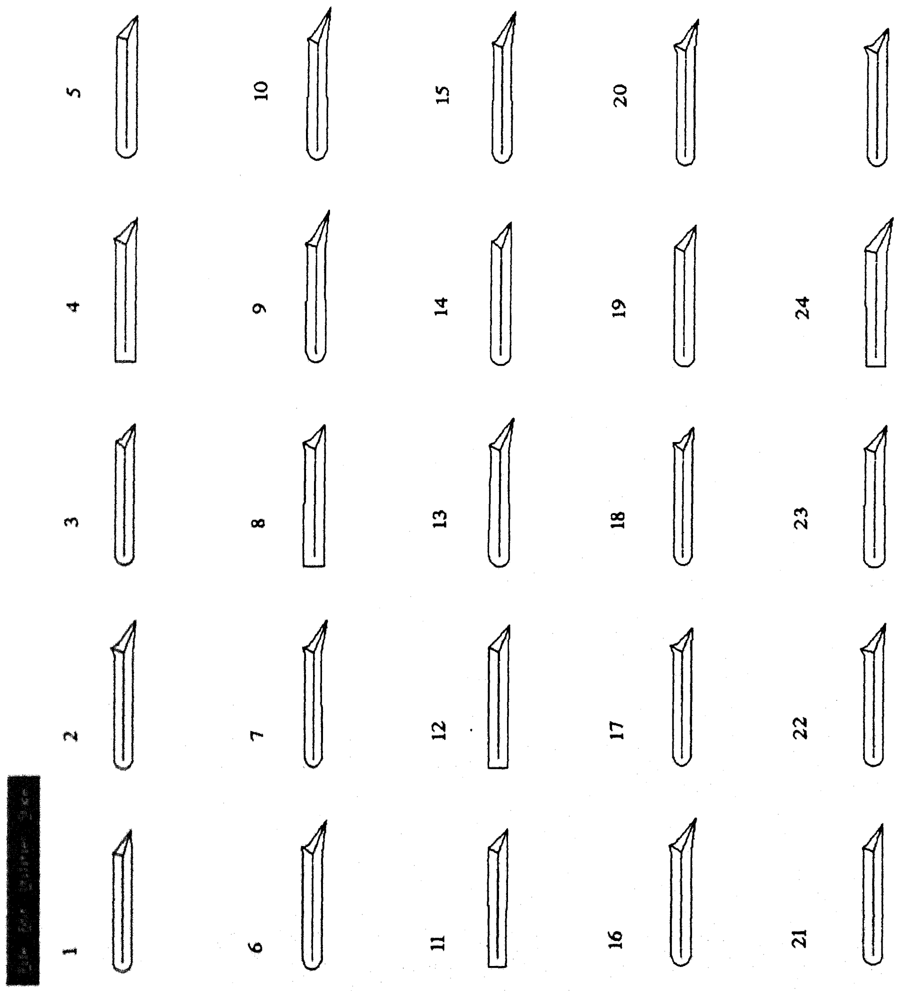


Figure 3.11: (a) *Knife to Chisel*: Generation 3. In the initial generation, each individual is formed by randomly choosing variables from within the limits shown in Fig.3.9. This population is obtained in generation 3 in the non-interactive mode.

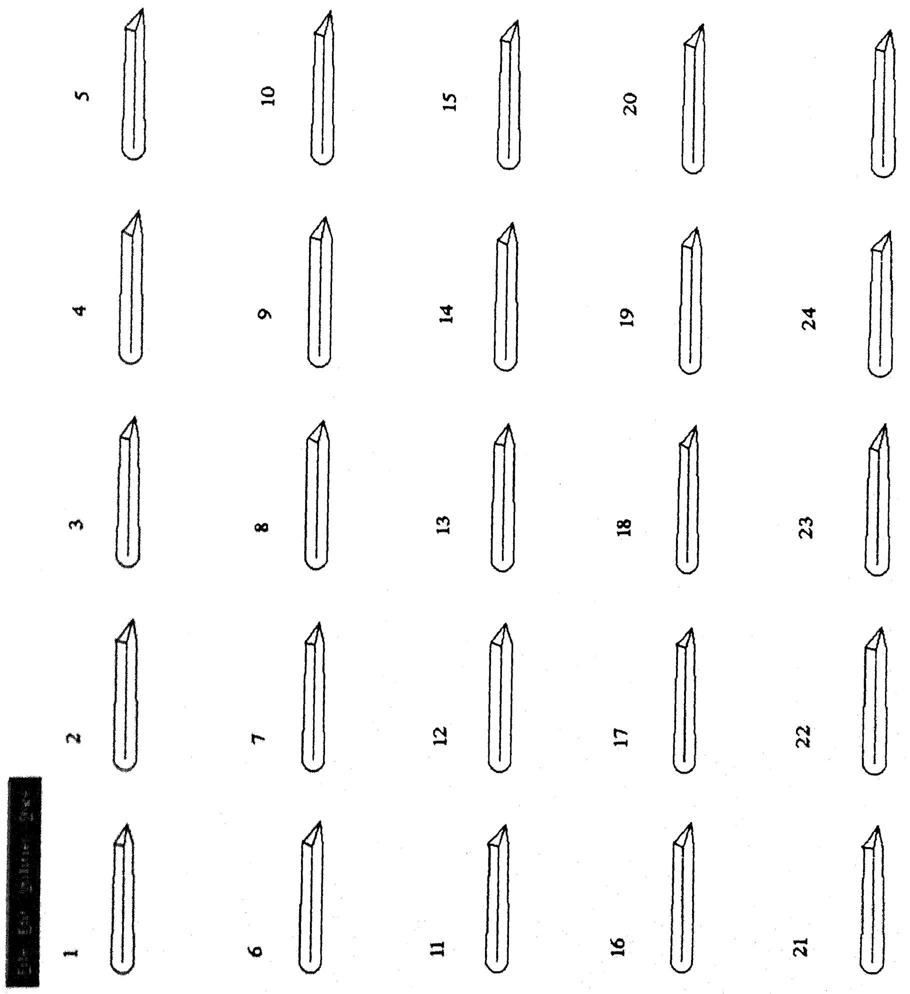


Figure 3.11: (b) *Knife to Chisel* : Generation 4. The population reflects a gradual transformation after the parameter ranges were redefined as shown in Table 3.2.

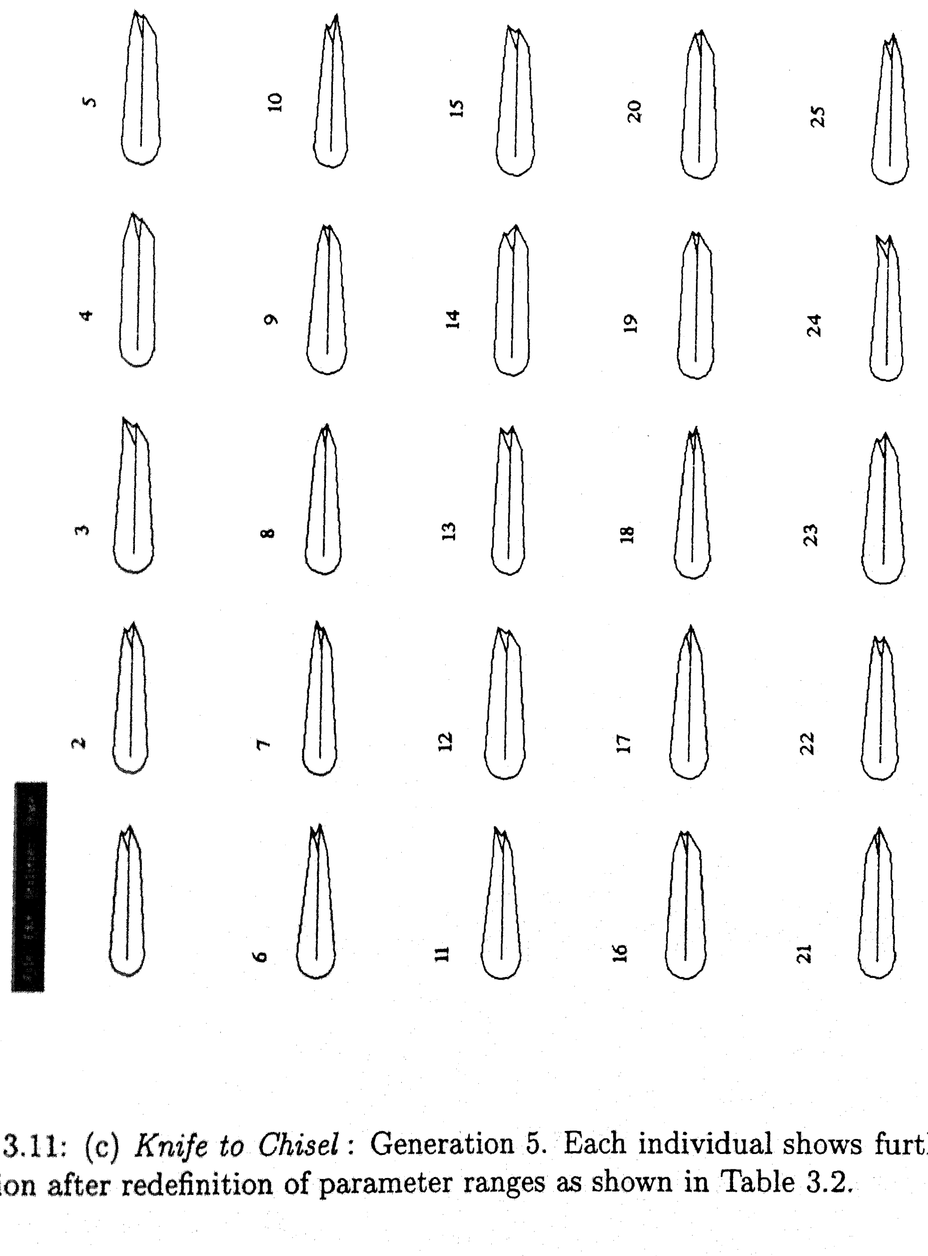


Figure 3.11: (c) *Knife to Chisel* : Generation 5. Each individual shows further transformation after redefinition of parameter ranges as shown in Table 3.2.

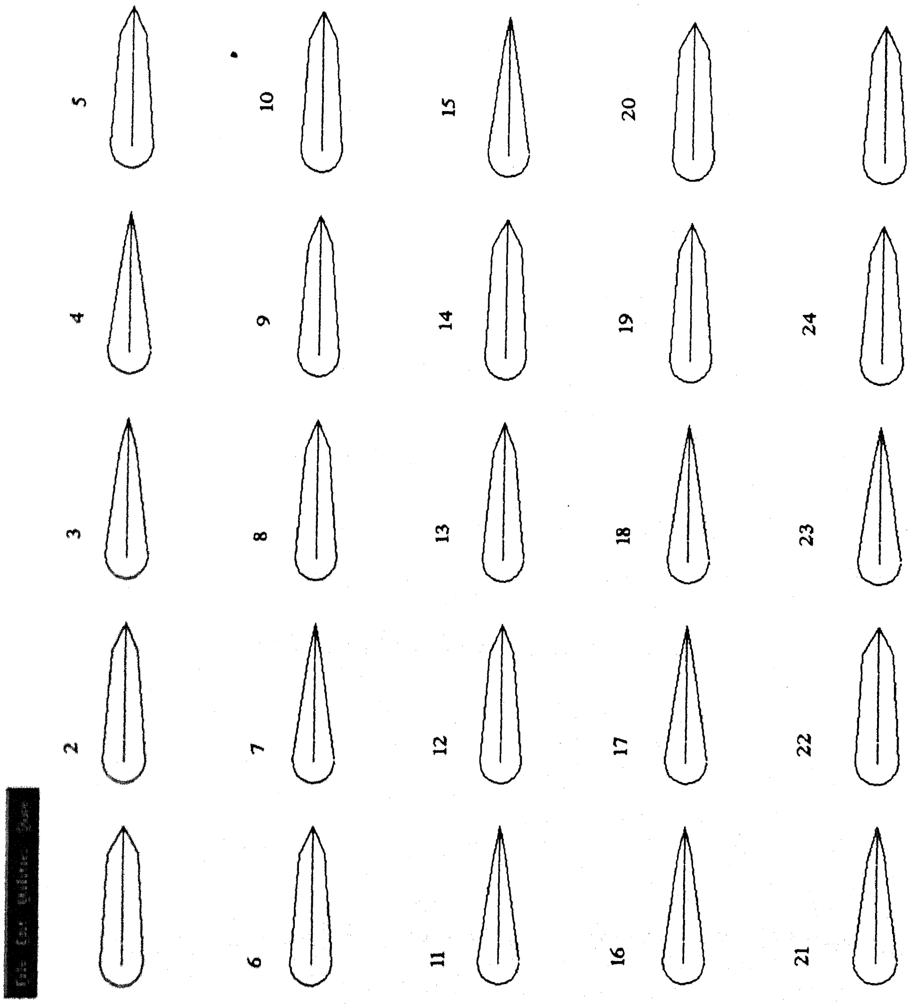


Figure 3.11: (d) *Knife to Chisel* : Generation 7. The population as a whole reflects near chisel shapes. This generation was obtained after modifying the parameter ranges further as shown in Table 3.2.

Variable	Gen 0	Gen 2	Gen 4	Gen 7
r_1	(0.2, 0.2)	(0.2, 0.2)	(0.2, 0.2)	(0.2, 0.2)
r_2	(0.2, 0.2)	(0.2, 0.2)	(0.2, 0.2)	(0.2, 0.2)
r_3	(0.05, 0.15)	(0.1, 0.2)	(0.2, 0.2)	(0.2, 0.2)
r_4	(0.2, 0.35)	(0.1, 0.2)	(0.03, 0.06)	(0.02, 0.02)
l_2	(0.2, 0.4)	(0.3, 0.4)	(0.3, 0.3)	(0.3, 0.3)
l_3	(0.6, 0.7)	(0.5, 0.6)	(0.5, 0.6)	(0.55, 0.55)
θ_1	(40, 50)	(60, 80)	(75, 90)	(90, 90)
θ_2	(60, 70)	(70, 90)	(90, 110)	(105, 110)

Table 3.3: QMAT data used for pistol to hammer transformation.

3.5.2 Pistol to Hammer

In another example, the transformation from a pistol shape to a hammer shape is carried out by following the same procedure (Fig.3.12 (a) through (d)). The set of data for Fig.3.12 (a) through (d) are shown in Table 3.3.

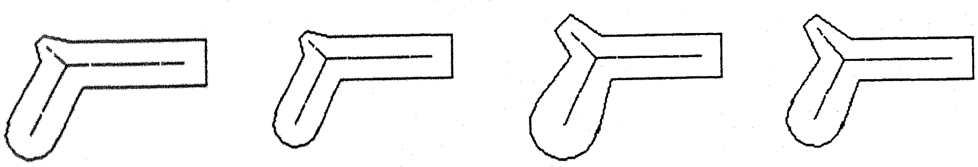


Figure 3.12: (a) *Pistol to Hammer*: Generation 2. Samples obtained using the variable ranges for Generation 0 shown in Table 3.3.

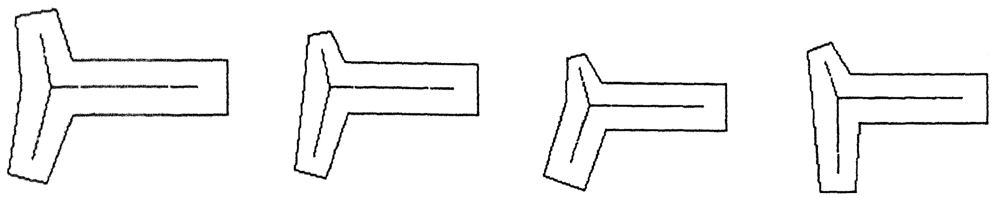


Figure 3.12: (b) *Pistol to Hammer* : Generation 4. Samples in this generation reflect a transformation away from the pistol shape. The parameters were redefined in Generation 2 after viewing the individuals as shown in Table 3.3.

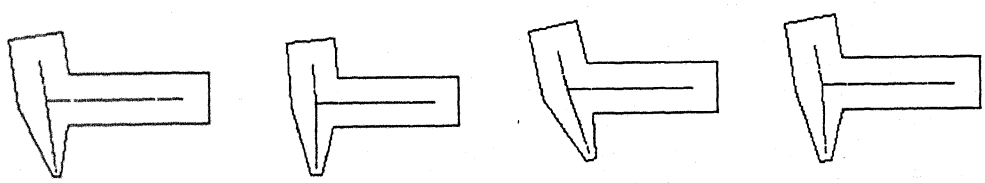


Figure 3.12: (c) *Pistol to Hammer* : Generation 7. Samples in this generation reflect a trend towards the hammer shape. The parameters were redefined in Generation 4 after viewing the samples as shown in Table 3.3.

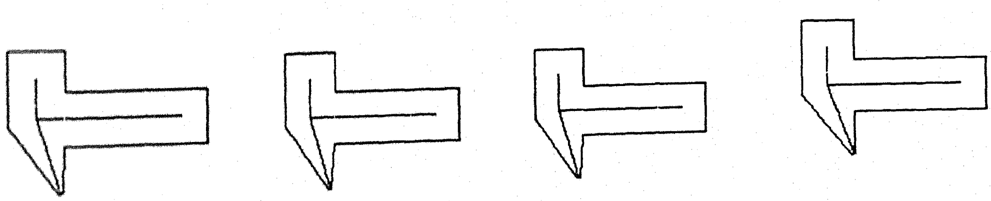


Figure 3.12: (d) *Pistol to Hammer* : Generation 10. Samples reflect a near hammer shape. The parameter ranges were modified again in Generation 7 after viewing the population as shown in Table 3.3.

3.5.3 Mercedes logo to Tata logo

The Mercedes logo is characterized by an inverted 3-arm star shape while the Tata logo is characterized by a T-shape. Fig.3.13 (a) through (e) traces this transformation. The data set for this example are shown in Table 3.4.

Table 3.4: QMAT data used for Mercedes logo to Tata logo.

Variable	Gen 0	Gen 2	Gen 4	Gen 7	Gen 10
r_1	(.015, .025)	(0.02, 0.06)	(0.08, 0.1)	(0.1, 0.15)	(0.15, 0.15)
r_2	(0.15, 0.25)	(0.15, 0.3)	(0.15, 0.2)	(0.15, 0.15)	(0.15, 0.15)
r_3	(.015, .025)	(0.02, 0.06)	(0.08, 0.1)	(0.1, 0.15)	(0.15, 0.15)
r_4	(.015, .025)	(0.02, 0.06)	(0.08, 0.1)	(0.1, 0.15)	(0.15, 0.15)
l_2	(0.9, 1.0)	(0.7, 1.0)	(0.6, 0.8)	(0.6, 0.7)	(0.6, 0.6)
l_3	(0.9, 1.0)	(0.7, 1.0)	(0.6, 0.8)	(0.6, 0.7)	(0.6, 0.6)
θ_1	(55, 65)	(50, 70)	(60, 80)	(75, 90)	(90, 90)
θ_2	(55, 65)	(50, 70)	(60, 80)	(75, 90)	(90, 90)

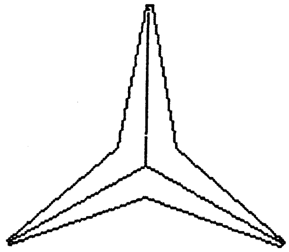


Figure 3.13: (a) *Mercedes logo to Tata logo* : Generation 2. The sample shows the Mercedes logo characterized by an inverted star. This sample was obtained using the variable ranges for Generation 0 as shown in Table 3.4.

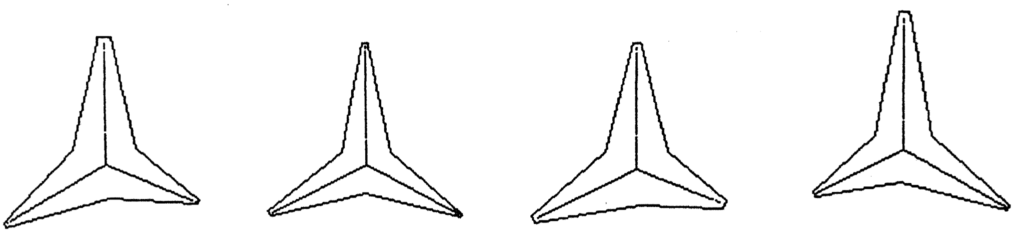


Figure 3.13: (b) *Mercedes logo to Tata logo* : Generation 4. Samples in this generation reflect a transformation away from the mercedes logo shape. The parameters were redefined in Generation 2 after viewing the samples shown in Fig.3.13(a).

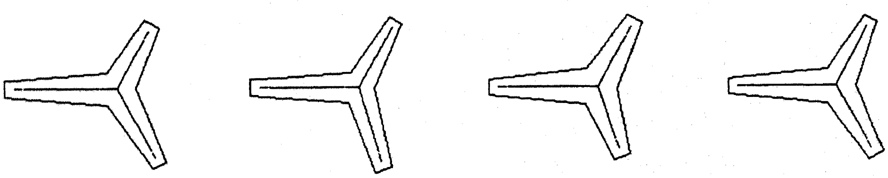


Figure 3.13: (c) *Mercedes logo to Tata logo* : Generation 7. Samples in this generation reflect further transformation away from the Mercedes logo shape. The parameters were redefined in Generation 4 as shown in Table 3.4.

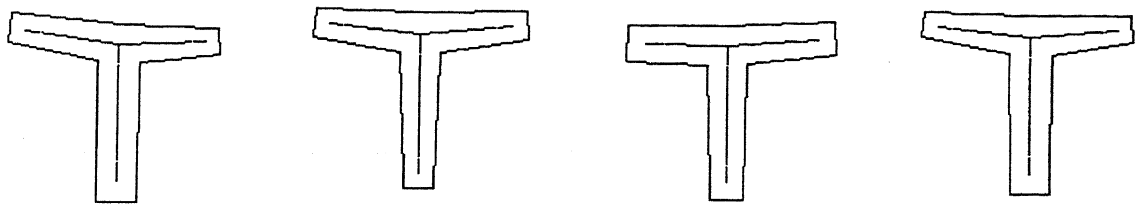


Figure 3.13: (d) *Mercedes logo to Tata logo* : Generation 10. The samples reflect near Tata logo shape (characterized by a T-shape) after the parameter ranges were modified again in Generation 7 as shown in Table 3.4.

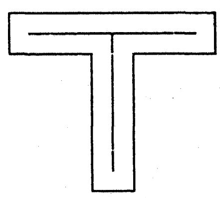


Figure 3.13: (e) The Tata logo obtained after fine tuning the parameter ranges in Generation 10 as shown in Table 3.4.

Chapter 4

Conclusions and Extensions

4.1 Conclusions

A serious shortcoming of today's CAD systems is the fact that precoded design knowledge does not allow designers to express their creative ideas. It does not allow design abstractions as inherent in design sketches. The major objective of this work has been representing and providing several modalities for leading the design from the initial ambiguous design space to increasingly well-defined shapes and finally to a single acceptable shape.

The representation of ambiguous shapes is based on an axial model of the shape. The basic technique can be used for sketch optimization, shape visualization, visual recognition, mental imagery etc. The system provides the user with a platform for interactive optimization. The user is shown a number of individuals from the design space rather than a few samples so that he is able to view the best with the worst along with the entire range in between. This dynamic visualization helps the user in emphasizing a particular trait, or for re-defining the design space itself. As shown in section 3.5, this can lead to a conceptually different design. More importantly, for the purpose of conceptual design, this enables the user to start with *sketchy* information rather than having to work out all the details before approaching a computer. The system now works with the user on the shape class embodied in his sketch and even allows the consideration of abstract constraints which can not be quantitatively defined ("Ah – just what I had in mind !" or just "I do not like this"), as is common in the

mental model of any designer during the earliest stages of the design process.

4.2 Extensions

The work done in this thesis opens up a number of avenues to be further explored.

4.2.1 Model for n-link case

The present model uses a 3-link MAT as the basic model which severely restricts the representation of more complex shapes. General shapes as shown in Fig.1.5 are too complex for 3-link MATs. Recently, work has been done towards generation of n-link MATs [12]. Fig.4.1 illustrates examples of more complex shapes for which MATs can be generated. This work uses a 3-link MAT for simplicity, but in practice, a more general shape is needed for dealing with real world applications.

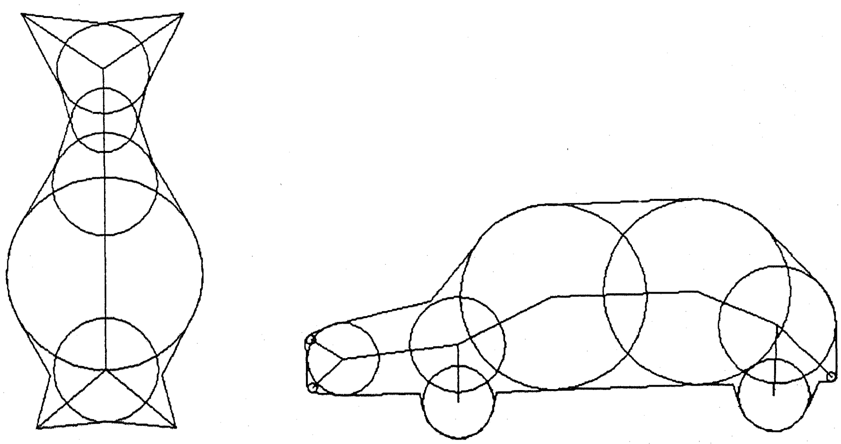


Figure 4.1: Some complex shapes along with their MATs (Taken from [12]).

4.2.2 Functional relations

In general, design can be seen as the transformation of functional requirements into a product which fulfills these requirements. In mechanical design, there is a wide spectrum of functions and a large number of ways to represent them. The development

of satisfactory descriptions of functions in a manner that is appropriate for computer solution and to capture and use the designer's intentions relating to function can be taken up as a further extension of this work.

4.2.3 Prototyping or Simulation based input

Another aspect of modeling function would be to enable the user to choose a figure and test it for some functional criterion using, say, an FEM program. Such a program can be integrated with the shape generation model and the performance in simulation used to determine the fitness value for each individual. With the advent of rapid prototyping, even actual tests may be possible on some of the samples.

4.2.4 Implementation in 3-D

Our model currently deals with 2-D shapes. It may be possible to extend this implementation to 3-D shapes especially for satisfying the functional requirements when dealing with real world applications (n-link cases). This can be done as an extrusion of the 2-D contour as in [6], or as a more general 3-D surface-site Voronoi diagram.

4.2.5 Further studies on interactive selection

In the direct selection mode, it should be investigated whether the user needs to select the shapes with some desired traits in each generation, or after k generations. Also, the number of individuals that must be clicked upon to achieve the desired characteristics remains to be explored. The effect of varying these factors on the results in different runs must be investigated.

4.2.6 Discovery

For large changes in the parameter redefinition mode, there is a need to redefine the SBX crossover function to generate a greater diversity in the population. The effect of varying the values of 'β' and 'n' needs to be further explored.

Appendix A

Guidelines to use SHAPE_OPTIM

The SHAPE_OPTIM program consists of the following files :

1. main.c : Main program.
2. rga_new.c : GA implementation code.
3. mat.c : Supplementary routines to generate MATs.
4. GA_DISPLAY.c : Code for graphical display on X-Windows platform.
5. click.c : Supplementary routines for graphical display.
6. defaults : Default values for some parameters.
7. input : Input file – optional, can be given any name.

The implementation uses a 3-link MAT case, where all the three links are meeting at a point as shown in Fig.A.1.

The shape is represented by the MAT as shown in Fig.A.1 and the radii at nodes 1, 2, 3, 4 designated by r_1, r_2, r_3, r_4. Position of nodes 3 and 4 is determined by the lengths l_2 and l_3, which are the lengths 2-3 and 2-4 respectively, taking length 1-2 as the reference link of unit length, and the angles θ_1 and θ_2, which are the angles 3-2-P and 4-2-P, respectively. Since these parameters are represented in a qualitative zone, the user is supposed to supply the uncertainity zones for each of these parameters to

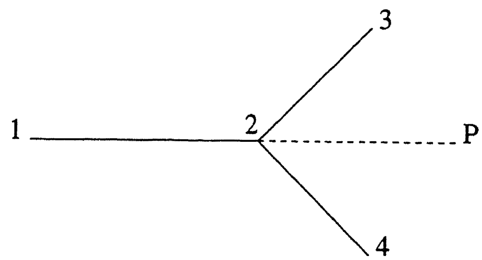


Figure A.1: Representation of a 3-link MAT

the GA program, for variables x[0] ..... x[7]. The variables are given in the following order :

r_1, r_2 ,r_3, r_4 ,l_2, l_3, θ_1, θ_2.

Check that the macro #define MAT_PROBLEM is defined in the header of rga_new.c. The SHAPE_OPTIM program is compiled and linked using a 'Makefile'. To run the program, give the following commands :

make : To compile and link all the files.

SHAPE_OPTIM : To execute the program.

At the execution of SHAPE_OPTIM command, the program asks the user if the data is to be input through a commented file. Give the answer 'n' (No) if the data input is through the keyboard and 'y' (Yes) if through a file. Then the program asks for the name of the input file to be supplied by the user. A sample of the input file used in the rocker arm example is shown below :

INPUT FILE FOR SHAPE_OPTIM :

How many generations ? ———- : 35

Population Size ? ——— : 25

Cross Over Probability ? (0 to 1) : 1.0

Mutation Probability ? (0 to 1) — : 0.01

Number of variables (Maximum 20) — : 8

Lower and Upper bounds of x[1] to x[N] :

0.2 0.32

0.35 0.45

0.2 0.32

0.1 0.2

0.8 1.0

0.7 0.8

-10 10

75 105

Are these bounds rigid ? (y/n) : y

Number of discrete variables ? : 0

String length ? : 30

Sharing ? (y/n) Sigma_share if yes : n

Reports to be printed ? (y/n) : y

Graphics work to be done ? (y/n) : n

How many runs ? : 1

random seed number : 0.1

Convergence and Closeness epsilons? : 1e-4 -1e-4

Maximum allowable spread of variables : 100.0

Enter tournament size for selection : 2

Give target value for x[1] to x[N] : 0 0 0 0 0 0 0 0

Target fitness value ? : -100

type of cross-over strategy.

1. xover in one variable

2. xover in all variables

3. xover on a straight line

Your choice : 2

Type of cross over ? (s/b) : s

Lower value and upper value of n ? : 2 2

(one can invoke any one or many penalty functions by giving some non-zero weight value corresponding to that penalty. Usually a value of 1.0 is preferable, but one can give any other value depending upon the requirement) :

0. Infeasibility : Removes infeasible shapes from consideration. Usually, this penalty function should be invoked necessarily.

1. Penetration : Removes those shapes which are near-circular type.

2. Non-convexity : Gives some penalty value to non-convex shapes, and prefers convex shapes. 3. Number of Edges : Prefers those shapes, which have less number of edges in their contour (simpler shapes).

4. Parallelism : If there are near-parallel edges in the contour, this penalty function tries to make them exactly parallel.

5. Aspect Ratio : Tries to generate shapes having aspect ratio within the given range. This range is supplied by the user in the input file.

6. Area (minimization) : Invoking this penalty function will make the shapes with lesser areas preferable over ones having large area. If one wants to do the opposite, i.e., area maximization, simply supply some negative value for the weight to this penalty.

7. Smoothness : Tries to generate more smooth shapes (the ones avoiding sharp changes in the angles of the boundary).

8. Perpendicularism : If there are near-perpendicular edges in the contour, this penalty function tries to make them exactly perpendicular.

In addition to these properties, which are being 'optimized', there is one more property which is specified i.e., linearity of the shape. There are two types of linearities :

1. Extended tangent linearity : If this is set to 1.0 all arcs will be approximated by two tangent lines.

2. Truncated tangent linearity : If this is set to 1.0 all arcs will be approximated by three tangent lines.

If these linearities are set to 0.0, the arcs will remain arcs. Any value between 0 to 1 will make some of the arcs to be linearized and rest of them to remain arcs.

In addition to these penalty functions, the user can specify the population size and the maximum number of generations.

To get full reports, give answer 'y' to the question "reports to be printed ?" This will generate the report for each generation giving the solution points, the objective fn. value and similar information.

WARNING : This may take lot of space while writing the file 'realga.out'

A turbo-C version of a graphical display program is available as "grap.c", which runs on DOS platform. To generate the datafiles for this, give answer 'y' to the question "graphics work to be done?". However, these files are not required for the interface in the SHAPE_OPTIM program and the user must give the answer 'n' to this query.

The file'defaults' contains the default values for some parameters. Refer [1] for detailed explanation of these parameters and other GA related parameters used in the input file.

During the run, for each generation, the program generates a file 'mat.dat' which is a datafile that stores the various shapes generated in the form of lines, circles and arcs. This file is thereafter used for the graphical display by the GA_DISPLAY.c routine.

To start with, the GA generates its initial population at random using the parameter ranges defined by the user. This population is then displayed on the screen for the user's perusal in a seperate window. Out of the shapes displayed, the user may see the best sample, zoom any individual, view the MAT or contour separately for any shape, save any shape individually (in the zoom/ best sample mode) or the entire population. These can be done by clicking on the buttons provided in the interface.

The interface also enables the user to emphasize certain characteristics in some shapes by clicking on them using the left mouse button. The reference for these chosen individuals is stored in the file 'click.dat' and read by the GA program rga_new.c, before generating the next population. The details have been discussed earlier in section 3.4.1. To redefine the parameter ranges while the run is on (as discussed in section 3.5), use the 'redefine-params' button in the 'Utilities' menu provided on the screen. The modified values for the desired parameters can be input through the root

window.

When the user is ready to proceed to the next generation, click on the 'Continue' button in the 'Utilities' menu. The user can quit the program using the 'quit' button in the 'File' menu. After completion, the program generates a file 'realga.out' which is a result file from the GA program and gives a summary of the results, in the context of optimization.

Limitations :

1. Only those samples which are liked by the user must be chosen. The converse does not hold good i.e., bad samples can not be discarded away.

2. The procedure requires a lot of memory space to accomodate the various files generated during the run and after completion.

References

[1] Agarwal, R.B. (1995). Simulated binary crossover for real- coded Genetic Algorithms : Development and application in Ambiguous Shape Modeling, Master's thesis, Dept. of Mech. Engg., IIT, Kanpur.

[2] Agarwal, R.B., Mukerjee, A., and Deb, K. (1995). Modelling of inexact 2-D shapes using real- coded genetic algorithms. In *Proceedings of the Symposium on Genetic Algorithms*, ed. Pradosh K. Roy and S.D. Mehta, Bishen Singh Mahendra Pal Singh publishe¿ rs, pages 41- 49.

[3] Biederman, I. (1990). Higher level vision. In *Visual cognition and action*, ed. Daniel N. Osherson et al, MIT Press, pages 41- 72.

[4] Chou, J.J. (1995). Voronoi diagram for planar shapes, In *IEEE Computer Graphics and Applications*, March 1995.

[5] Coyne, R.D., M.A. Rosenman, A.D. Radford, M. Balachandran, and J.S. Gero, 1990, Knowledge- based design systems, Addison Wesley, 1990.

[6] King, J.S., and Mukerjee, A. (1990). Inexact Visualization, In *Proceedings of the IEEE Conference on Biomedical Visualization*, Atlanta GA, May 22-25, 1990, pages 136-143.

[7] King, J.S. (1991). Inexact Visualization : Qualitative shape representation for recognizable reconstruction, Master's thesis, Dept. of Comp. Sc., Texas A and M Univ.

[8] Mittal, N., and Mukerjee, A. (1995). Qualitative subdivision algebra : Moving towards the quantitative. IIT Kanpur, Department of Mechanical Engg, Technical report ME-95-011, and UT Austin Computer Science Tech report 95-23, 16 pages. [ftp://ftp.cs.utexas.edu/pub/techreports/tr95-23.ps.Z]

[9] Mukerjee, A., and Joe, G. (1990). A qualitative model for space. In 8th – AAAI (1990), pages 721- 727.

[10] Mukerjee, A. (1990). Qualitative Geometric Design. In *Solid Modeling Foundations and CAD/CAM Applications*, ed. J.R. Rossignac and J. Turner, Springer Verlag 1991.

[11] Mukerjee, A. (1995). Querying Spatial Features by Shape. Internal document. [ftp://ftp.cs.albany.edu/pub/amit/shapegis.ps.gz]

[12] Mukerjee, A., R.B. Agarwal, N. Tiwari, and N. Hasan. (1996). Qualitative Sketch Optimization. *J. of AI in Engineering Design and Manufacturing*, (to appear).

[13] Requicha, A.A.G. (1980). Representation for Rigid Solids : Theory, Methods, and systems, ACM computer surveys, Dec. 1980.

Date Slip

This book is to be returned on the date last stamped.
